Algorithmes

Yann Collette - Patrick Siarry

Optimisation multiobjectif

EYROLLES

Optimisation multiobjectif

DANS LA MÊME COLLECTION

G. DREYFUS *et al*. – **Réseaux de neurones**.
Méthodologie et applications.
N°11019, 2002, 380 pages.

A. CORNUÉJOLS, L. MICLET. – **Apprentissage artificiel.**
Concepts et algorithmes.
N°11020, 2002, 638 pages.

C. GUÉRET, C. PRINS, M. SEVAUX. – **Programmation linéaire.**
*65 problèmes d'optimisation modélisés et résolus avec Visual XPress***.**
N°9202, 2000, 365 pages, *avec CD-ROM.*

CHEZ LE MÊME ÉDITEUR

M. GONDRAN, M. MINOUX. – **Graphes et algorithmes.**
N°1571, 1995, 622 pages.

A. CARDON. – **Conscience artificielle et systèmes adaptatifs**.
N°9124, 2000, 384 pages.

W. MINKER, S. BENNACEF. – **Parole et dialogue homme-machine**.
N°5824, 2000, 212 pages.

BOURDA. – **Introduction à l'informatique théorique**.
N°1642, 1994, 236 pages.

Yann Collette - Patrick Siarry

Optimisation multiobjectif

EYROLLES

ÉDITIONS EYROLLES
61, Bld Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com



Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissement d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'Éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.

© Groupe Eyrolles, 2002, ISBN 2-212-11168-1

Optimisation Multiobjectif

Yann Collette[1], Patrick Siarry[2]

1 Professeur agrégé,
ancien élève de l'ENS de Cachan
doctorant à
Electricité de France,Clamart

2 Professeur à l'Université de
Paris XII Val-de-Marne

Table des matières

Avant-propos

Les ingénieurs se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers, comme dans le traitement des images, la conception de systèmes mécaniques, la recherche opérationnelle et l'électronique. Le problème à résoudre peut fréquemment être exprimé sous la forme générale d'un *problème d'optimisation*, dans lequel on définit une fonction objectif, ou fonction de coût (voire plusieurs), que l'on cherche à minimiser par rapport à tous les paramètres concernés. Par exemple, dans le célèbre problème du voyageur de commerce, on cherche à minimiser la longueur de la tournée d'un "voyageur de commerce", qui doit visiter un certain nombre de villes, avant de retourner à la ville de départ. La définition du problème d'optimisation est souvent complétée par la donnée de *contraintes* : tous les paramètres (ou *variables de décision*) de la solution proposée doivent respecter ces contraintes, faute de quoi la solution n'est pas réalisable.

Il existe de nombreuses méthodes d'optimisation "classiques" pour résoudre de tels problèmes, applicables lorsque certaines conditions mathématiques sont satisfaites : ainsi, la *programmation linéaire* traite efficacement le cas où la fonction objectif, ainsi que les contraintes, s'expriment linéairement en fonction des variables de décision. Malheureusement, les situations rencontrées en pratique comportent souvent une ou plusieurs complications, qui mettent en défaut ces méthodes : par exemple, la fonction objectif peut être non linéaire, ou même ne pas s'exprimer analytiquement en fonction des paramètres ; ou encore, le problème peut exiger la considération simultanée de plusieurs objectifs contradictoires.

Pour clarifier l'analyse, on peut s'intéresser séparément à deux domaines, l'optimisation *multiobjectif* et l'*optimisation difficile*, qui mettent l'accent sur deux sources différentes de complications. Le premier domaine, auquel cet ouvrage est consacré, traite le cas de la présence simultanée de plusieurs objectifs, tandis que le second analyse les difficultés inhérentes aux propriétés mêmes de chaque fonction objectif. Nous verrons que les deux aspects sont souvent liés en pratique. Avant de se focaliser sur l'optimisation multiobjectif, il est nécessaire d'expliciter maintenant davantage le cadre de l'optimisation difficile.

Optimisation difficile

On distingue en réalité deux types de problèmes d'optimisation : les problèmes combinatoires (ou problèmes "discrets") et les problèmes à variables continues. Pour clarifier, citons quelques exemples : parmi les problèmes combinatoires, on trouve le problème du voyageur de commerce mentionné plus haut, ou encore celui de l'ordonnancement de tâches concourant à un même projet. Deux exemples de problèmes continus sont les suivants : l'identification des paramètres d'un modèle associé à un composant électronique, à partir de données expérimentales caractérisant celui-ci ; la restauration optimale d'une image, à partir d'une image brouillée.

Cette différenciation est nécessaire pour cerner le domaine de l'optimisation difficile. En effet, deux sortes de problèmes reçoivent, dans la littérature, cette appellation, non définie strictement (et liée, en fait, à l'état de l'art en matière d'optimisation) :

- certains problèmes d'optimisation combinatoire, pour lesquels on ne connaît pas d'algorithme exact *rapide*. C'est le cas, en particulier, des problèmes dits "NP-difficiles", c'est-à-dire - schématiquement - dont la résolution exacte n'est pas possible en un temps de calcul proportionnel à N^n, où N désigne le nombre de paramètres inconnus du problème, et n est un entier.
- certains problèmes d'optimisation à variables continues, pour lesquels on ne connaît pas d'algorithme permettant de repérer un optimum *global* à coup sûr et en un nombre fini de calculs.

Des efforts ont longtemps été menés, séparément, pour résoudre ces deux types de problèmes "difficiles". Dans le domaine de l'optimisation continue, il existe ainsi un arsenal important de méthodes classiques dites d'*optimisation globale*, mais ces techniques sont souvent inefficaces si la fonction objectif ne possède pas une propriété structurelle particulière, telle que la convexité. Dans le domaine de l'optimisation combinatoire, un grand nombre d'*heuristiques*, qui produisent des solutions proches de l'optimum, ont été développées ; mais la plupart d'entre elles ont été conçues spécifiquement pour un problème donné.

Métaheuristiques

L'arrivée d'une nouvelle classe de méthodes, nommées *métaheuristiques*, marque une réconciliation des deux domaines : en effet, celles-ci s'appliquent à toutes sortes de problèmes combinatoires, et elles peuvent également s'adapter aux problèmes continus. Ces méthodes, qui comprennent notamment la méthode du recuit simulé, les algorithmes génétiques, la méthode de recherche tabou, les algorithmes de colonies de fourmis, etc. sont apparues, à partir des années 1980, avec une ambition commune : résoudre *au mieux* les problèmes d'optimisation difficile. Elles ont en commun, en outre, les caractéristiques suivantes :

- elles sont, au moins pour partie, *stochastiques* : cette approche permet de faire face à l'*explosion combinatoire* des possibilités ;
- elles sont d'origine combinatoire : elles ont l'avantage, décisif dans le cas continu, d'être *directes*, c'est-à-dire qu'elles ne recourent pas au calcul, souvent problématique, des gradients de la fonction objectif ;
- elles sont inspirées par des *analogies* : avec la physique (recuit simulé, diffusion simulée, etc.), avec la biologie (algorithmes génétiques, recherche tabou, etc.) ou avec l'éthologie (colonies de fourmis, essaims de particules, etc.) ;
- elles sont capables de guider, dans une tâche particulière, une autre méthode de recherche spécialisée (par exemple, une autre heuristique, ou une méthode d'exploration *locale*) ;
- elles partagent aussi les mêmes inconvénients : les difficultés de *réglage* des paramètres mêmes de la méthode, et le *temps de calcul* élevé.

Les métaheuristiques ne s'excluent pas mutuellement : en effet, dans l'état actuel de la recherche, il est le plus souvent impossible de prévoir avec certitude l'efficacité d'une méthode donnée, quand elle est appliquée à un problème donné. De plus, la tendance actuelle est l'émergence de *méthodes hybrides*, qui s'efforcent de tirer parti des avantages spécifiques d'approches différentes en les combinant. Nous verrons que les métaheuristiques jouent un rôle important dans cet ouvrage, car elles ont largement contribué au renouvellement de l'optimisation multiobjectif.

Il nous paraît intéressant, pour clore ces considérations préliminaires sur l'optimisation monobjectif difficile, d'analyser brièvement la source de l'efficacité des métaheuristiques, puis de souligner quelques unes des extensions de ces méthodes. Cette étude nous permettra d'esquisser une classification générale des méthodes d'optimisation monobjectif.

Source de l'efficacité des métaheuristiques

Pour faciliter l'exposé, prenons un exemple simple de problème d'optimisation : celui du placement des composants d'un circuit électronique. La fonction objectif à minimiser est la longueur des connexions, et les variables de décision sont les emplacements des composants du circuit. L'allure de la fonction objectif de ce problème peut être schématiquement représentée comme sur la figure 1, en fonction de la "configuration" : chaque configuration est un placement particulier, associé à un choix de valeur pour chacune des variables de décision.

Lorsque l'espace des configurations possibles présente une structure aussi tourmentée, il est difficile de repérer le minimum global c_*. Nous expliquons ci-dessous l'échec d'un algorithme itératif "classique", avant de commenter la démarche fructueuse des métaheuristiques.

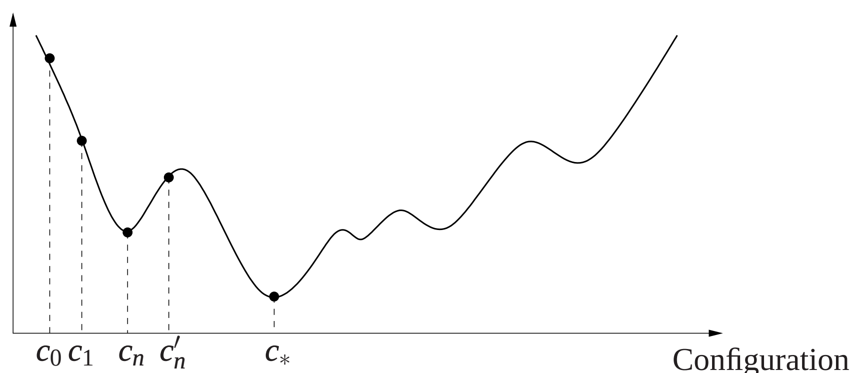


FIG. 1 – Allure de la fonction objectif d'un problème d'optimisation difficile en fonction de la "configuration".

Piégeage d'un algorithme itératif "classique" dans un minimum local

Le principe d'un algorithme classique d' "amélioration itérative" est le suivant : on part d'une configuration initiale c_0, qui peut être choisie au hasard, ou bien - par exemple dans le cas du placement d'un circuit électronique - qui peut être celle d'un concepteur. On essaie alors une modification élémentaire, souvent appelée "*mouvement*" (par exemple, on permute deux composants choisis au hasard, ou bien on translate l'un d'entre eux), et l'on compare les valeurs de la fonction objectif, avant et après cette modification. Si le changement conduit à une diminution de la fonction objectif, il est accepté, et la configuration c_1 obtenue, qui est "*voisine*" de la précédente, sert de point de départ pour un nouvel essai. Dans le cas contraire, on revient à la configuration précédente avant de faire une autre tentative. Le processus est itéré jusqu'à ce que toute modification rende le résultat moins bon. La figure 1 montre que cet algorithme d'amélioration itérative (désigné aussi sous les termes de *méthode de descente*) ne conduit pas, en général, au minimum absolu, mais seulement à un minimum local c_n, qui constitue la meilleure des solutions accessibles compte tenu de l'hypothèse initiale.

Pour améliorer l'efficacité de la méthode, on peut, bien entendu, l'appliquer plusieurs fois, avec des conditions initiales différentes choisies arbitrairement, et retenir comme solution finale le meilleur des minima locaux obtenus ; cependant, cette procédure augmente sensiblement le temps de calcul de l'algorithme, et ne garantit pas de trouver la configuration optimale c_*.

Capacité des métaheuristiques à s'extraire d'un minimum local

Pour surmonter l'obstacle des minima locaux, une autre idée s'est montrée très fructueuse, au point qu'elle est à la base de toutes les métaheuristiques dites ***de voisinage*** (recuit simulé, méthode tabou) : il s'agit d'autoriser, de temps en temps, des mouvements ***de remontée***, autrement dit d'accepter une dégradation temporaire de la situation, lors du changement de la configuration courante. C'est le cas si l'on passe de c_n à $c_n^{'}$ (figure 1). Un mécanisme de contrôle des dégradations - spécifique à chaque métaheuristique - permet d'éviter la divergence du procédé. Il devient, dès lors, possible de s'extraire du piège que représente un minimum local, pour partir explorer une autre "vallée" plus prometteuse.

Les métaheuristiques "distribuées" (telles que les algorithmes génétiques) ont elles aussi des mécanismes permettant la sortie hors d'un "puits" local de la fonction objectif. Ces mécanismes (comme la ***mutation*** dans les algorithmes génétiques) affectant une solution viennent, dans ce cas, seconder le mécanisme collectif de lutte contre les minima locaux, que représente le contrôle en parallèle de toute une "population" de solutions.

Extensions des métaheuristiques

Nous passons en revue quelques-unes des extensions qui ont été proposées pour faire face à des particularités de l'optimisation.

Adaptation aux problèmes à variables continues

Ces problèmes, de loin les plus courants en ingénierie, intéressent moins les informaticiens. La plupart des métaheuristiques, d'origine combinatoire, ont été cependant adaptées au cas continu, ce qui suppose notamment le recours à une stratégie de discrétisation des variables : le pas de discrétisation doit s'adapter en cours d'optimisation, pour garantir à la fois la régularité de la progression vers l'optimum et la précision du résultat.

Parallélisation

De multiples modes de parallélisation ont été proposés pour les différentes métaheuristiques. Certaines techniques se veulent générales ; d'autres, en revanche, tirent avantage des particularités du problème. Ainsi, dans les problèmes de placement de composants, les tâches peuvent être réparties naturellement entre plusieurs processeurs : chacun d'eux est chargé d'optimiser une zone géographique donnée et des informations sont échangées périodiquement entre processeurs voisins.

Optimisation multimodale

Il s'agit cette fois de déterminer tout un jeu de solutions optimales, au lieu d'un optimum unique. Les algorithmes génétiques sont particulièrement bien adaptés à cette tâche, de par leur nature distribuée. Les variantes de type "multipopulation" exploitent en parallèle plusieurs populations, qui s'attachent à repérer des optima différents.

Les méthodes hybrides

Après le triomphalisme des débuts des tenants de telle ou telle métaheuristique, l'heure est venue de faire un bilan réaliste et d'accepter la complémentarité de ces nouvelles méthodes entre elles, ainsi qu'avec d'autres approches : d'où l'émergence actuelle de *méthodes hybrides*.

Classification générale des méthodes d'optimisation monobjectif

Pour tenter de récapituler les considérations précédentes, nous proposons à la figure 2 une classification générale des méthodes d'optimisation monobjectif.

On retrouve, dans ce graphe, les principales distinctions opérées plus haut :

- On différencie, en premier lieu, l'optimisation combinatoire de l'optimisation continue.
- Pour l'optimisation combinatoire, on a recours aux méthodes approchées lorsqu'on est confronté à un problème difficile ; dans ce cas, le choix est parfois possible entre une heuristique "spécialisée", entièrement dédiée au problème considéré, et une métaheuristique.
- Pour l'optimisation continue, on sépare sommairement le cas linéaire (qui relève notamment de la *programmation linéaire*) du cas non linéaire, où l'on retrouve le cadre de l'optimisation difficile ; dans ce cas, une solution pragmatique peut être de recourir à l'application répétée d'une méthode locale qui exploite, ou non, les gradients de la fonction objectif. Si le nombre de minima locaux est très élevé, le recours à une méthode globale s'impose : on retrouve alors les métaheuristiques, qui offrent une alternative aux méthodes classiques d'optimisation globale, celles-ci requérant des propriétés mathématiques restrictives de la fonction objectif.
- Parmi les métaheuristiques, on peut différencier les métaheuristiques "de voisinage", qui font progresser une seule solution à la fois (recuit simulé, recherche tabou, etc.), et les métaheuristiques "distribuées", qui manipulent en parallèle toute une population de solutions (algorithmes génétiques, etc.).
- Enfin, les méthodes hybrides associent souvent une métaheuristique et une méthode locale. Cette coopération peut prendre la simple forme d'un passage de relais entre la métaheuristique et la technique locale, chargée d'affiner la solution. Mais les deux approches peuvent aussi être entremêlées de manière plus complexe.

Citons quelques exemples de méthodes illustrant ces différentes catégories :

Méthode exacte : cette méthode va rechercher l'optimum global. Pour cela, elle effectue, en général, une énumération des solutions de l'espace de recherche. Par exemple la méthode de "Branch and Bound" (voir [Hillier et al. 95]) dans laquelle on effectue les étapes suivantes :

- on divise l'espace de recherche en sous-espaces,
- on cherche une borne minimale en terme de fonction objectif associée à chaque sous-espace de recherche,
- on élimine les "mauvais" sous-espaces,
- on reproduit les étapes précédentes jusqu'à obtention de l'optimum global.

Cette méthode permet d'obtenir l'optimum global de manière exacte, sur certains types de problèmes. Il n'est pas possible d'appliquer cette méthode à des problèmes dont l'espace de recherche est de taille trop importante.

Méthodes approchées :

- Méthodes Heuristiques : elles sont développées pour résoudre un type de problème en particulier. Elles ne fonctionnent que sur le type de problème pour lequel elles ont été développées, par exemple (voir [Ausiello et al. 99]) l'algorithme de type "First Fit" dédié au problème de "Bin packing" (ou problème de rangement).
- Méthodes Métaheuristiques : elles sont fondées sur une idée générale (le parallèle avec un processus naturel, comme le recuit d'un métal, dans le cas du recuit simulé). Elles peuvent s'adapter à n'importe quel type de problème et donnent de bons résultats. Par exemple : le recuit simulé ou les algorithmes génétiques, qui seront présentés plus loin dans cet ouvrage.

Méthodes non-linéaires :

- Globale :

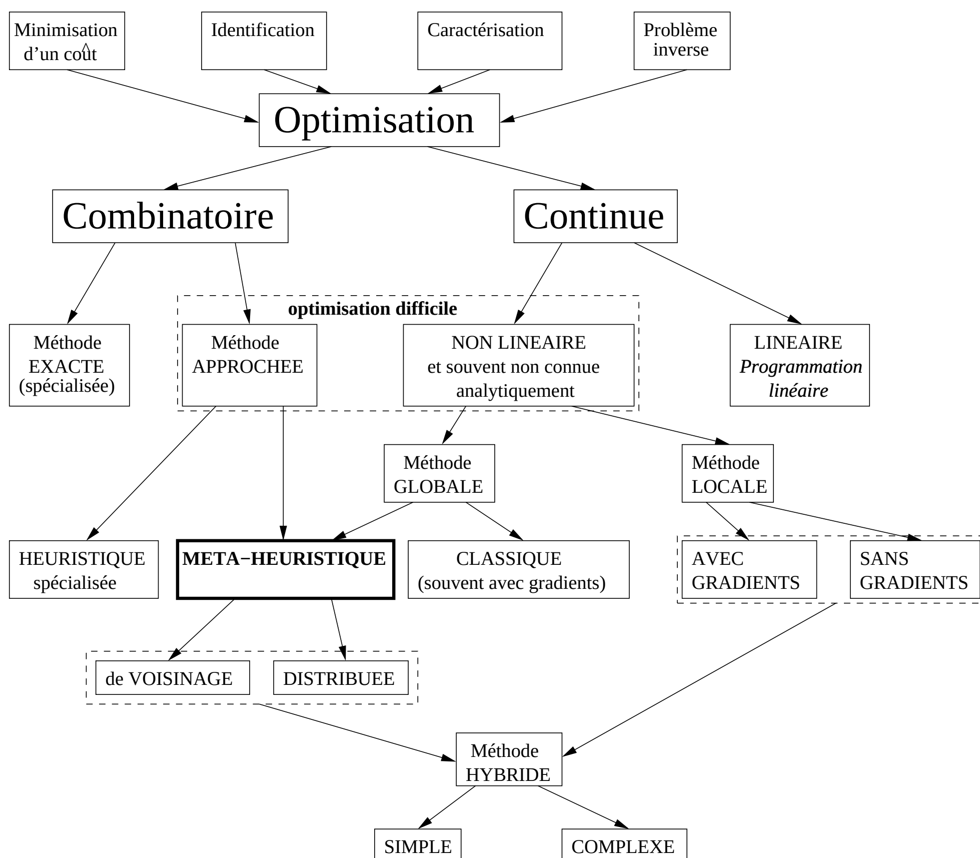


FIG. 2 – Classification générale des méthodes d'optimisation monobjectif.

  - Métaheuristiques : l'idée directrice sur laquelle repose une métaheuristique est suffisamment générale pour être transposable facilement aux problèmes d'optimisation continus.
  - Classiques : ces méthodes peuvent être considérées comme les méthodes "de base" de l'optimisation. Dans cette famille, on trouve la méthode de descente du gradient. Cette méthode, quand elle est appliquée à une fonction convexe, permet de trouver un optimum global.
- Locale :
  - Avec gradient : ces méthodes exploitent le gradient d'une fonction pour effectuer la recherche de l'optimum. Quand cette technique est appliquée à un problème d'optimisation continue de forme générale, il n'est en général pas possible de trouver l'optimum global du problème. On obtient alors un optimum local.
  - Sans gradient : cette famille regroupe une grande diversité de méthodes. Parmi ces méthodes, on peut citer :
    - Les méthodes constructives : on construit la solution une variable après l'autre, en interdisant de modifier une variable déjà affectée. Par exemple, la méthode glouton fait partie des méthodes constructives.

- Les méthodes stochastiques : elles reposent sur un processus aléatoire. Par exemple, on peut citer la méthode de marche aléatoire. Dans cette méthode, on tire un point situé dans le voisinage d'un point courant et ce point devient alors le point courant, quelle que soit la valeur de la fonction objectif.

Optimisation multiobjectif

La principale difficulté que l'on rencontre en optimisation monobjectif vient du fait que modéliser un problème sous la forme d'une équation unique peut être une tâche difficile. Avoir comme but de se ramener à une seule fonction objectif peut aussi biaiser la modélisation.

L'optimisation multiobjectif autorise ces degrés de liberté qui manquaient en optimisation monobjectif. Cette souplesse n'est pas sans conséquences sur la démarche à suivre pour chercher un optimum à notre problème enfin modélisé. La recherche ne nous donnera plus une solution unique mais une multitude de solutions. Ces solutions sont appelées *solutions de Pareto* et l'ensemble de solutions que l'on obtient à la fin de la recherche est la *surface de compromis*.

C'est après avoir trouvé les solutions du problème multiobjectif que d'autres difficultés surviennent : il faut sélectionner une solution dans cet ensemble. La solution qui sera choisie par l'utilisateur va refléter les compromis opérés par le décideur vis-à-vis des différentes fonctions objectif.

Le décideur étant "humain", il va faire des choix et l'un des buts de l'optimisation multiobjectif va être de modéliser les choix du décideur ou plutôt ses préférences. Pour modéliser ces choix, on pourra s'appuyer sur deux grandes théories :

- la théorie de l'utilité multi-attribut (voir [Keeney et al. 93]) ;
- la théorie de l'aide à la décision (voir [Roy et al. 93]).

Ces deux théories ont des approches différentes.

La première approche va chercher à modéliser les préférences du décideur, en postulant qu'implicitement chaque décideur cherche à maximiser une fonction appelée *fonction d'utilité*. Cette approche n'accepte pas qu'il y ait, en fin de sélection, des solutions ex-aequo.

La seconde approche, elle, va chercher à reproduire le processus de sélection du ou des décideurs. Pour cela, on s'appuiera sur des outils permettant d'opérer des sélections de solutions parmi un ensemble de solutions. Cette approche accepte que l'on obtienne des solutions ex-aequo.

L'autre difficulté à laquelle on sera confronté avant d'effectuer une optimisation sera la sélection de la méthode d'optimisation. En effet, on peut répartir les méthodes d'optimisation multiobjectif en trois grandes familles. Ces familles sont définies en fonction de l'instant où l'on prend la décision d'effectuer ou non un compromis entre fonctions objectif. Nous avons les familles suivantes :

- Les méthodes d'optimisation *a priori* :
  Dans ces méthodes, le compromis que l'on désire effectuer entre les différentes fonctions objectif a été déterminé avant l'exécution de la méthode d'optimisation. Pour obtenir la solution de notre problème, il suffira de faire une et une seule recherche et, en fin d'optimisation, la solution obtenue reflètera le compromis que l'on désirait effectuer entre les fonctions objectif avant de lancer la recherche.
  Cette méthode est intéressante dans le sens où il suffit d'une seule recherche pour trouver la solution. Cependant, il ne faut pas oublier le travail de modélisation du compromis qui a été effectué avant d'obtenir ce résultat. Il ne faut pas, non plus, oublier que le

décideur est “humain” et qu’il sera susceptible de constater que la solution obtenue en fin d’optimisation ne le satisfait finalement pas et qu’il souhaite maintenant, après avoir examiné cette solution, une solution qui opère un autre compromis entre les fonctions objectif.

– Les méthodes d’optimisation *progressive* :
Dans ces méthodes, on cherchera à questionner le décideur au cours de l’optimisation afin que celui-ci puisse réorienter la recherche vers des zones susceptibles de contenir des solutions qui satisfassent les compromis qu’il souhaite opérer entre les fonctions objectif.
Ces méthodes, bien qu’appliquant une technique originale pour modéliser les préférences du décideur, présentent l’inconvénient de monopoliser l’attention du décideur tout au long de l’optimisation.
Cet inconvénient n’est pas majeur lorsque l’on a affaire à des problèmes d’optimisation pour lesquels la durée d’une évaluation de la fonction objectif n’est pas importante. Pour les autres problèmes, l’utilisation d’une telle méthode peut être délicate. Il est en effet difficile de monopoliser l’attention du décideur pendant une période qui peut s’étendre sur plusieurs heures en lui posant, de plus, des questions toutes les dix minutes, voire même toutes les heures.
– Les méthodes *a posteriori* :
Dans ces méthodes, on va chercher un ensemble de solutions bien réparties dans l’espace de solutions. Le but sera ensuite de proposer ces solutions au décideur pour qu’il puisse sélectionner la solution qui le satisfait le plus en jugeant les différentes solutions proposées.
Ici, il n’est plus nécessaire de modéliser les préférences du décideur. On se contente de produire un ensemble de solutions que l’on transmettra au décideur. Il y a donc un gain de temps non négligeable vis-à-vis de la phase de modélisation des préférences de la famille des méthodes *a priori*.
L’inconvénient qu’il nous faut souligner est que, maintenant, il faut générer un ensemble de solutions bien réparties. Cette tâche est non seulement difficile, mais, en plus, peut requérir un temps d’exécution prohibitif.

Par rapport à l’optimisation monobjectif, on remarque que l’on a gagné des degrés de liberté supplémentaires pour modéliser notre problème. Par contre, alors que la sélection d’une méthode d’optimisation monobjectif était déjà difficile du fait de la diversité des méthodes d’optimisation disponibles, nous nous trouvons, en optimisation multiobjectif, face à une difficulté encore plus importante.

Ce nouveau type d’optimisation a cependant permis de résoudre avec succès de nombreux problèmes d’optimisation :
– dimensionnement de réseaux,
– optimisation de structure,
– problèmes d’ordonnancement,
– etc.

Le dynamisme de la communauté de l’optimisation multiobjectif montre que le domaine est en pleine expansion :
– de nouvelles conférences dédiées à l’optimisation multiobjectif voient le jour ;
– le nombre de publications relatives à l’optimisation multiobjectif croît rapidement ;
– l’intérêt porté à ce domaine par les industriels est désormais patent.

Dans cet ouvrage, nous présentons un panorama détaillé, mais non exhaustif, des méthodes d’optimisation multiobjectif et de la théorie s’y rapportant, et nous présentons quelques applications.

Plan de l'ouvrage

Partie I : Principe des méthodes d'optimisation multiobjectif

Chapitre 1 : Introduction : optimisation multiobjectif et dominance.

Nous exposons les différentes notions relatives à l'optimisation multiobjectif à travers des exemples simples. La lecture de ce chapitre clé est indispensable à la lecture du reste de l'ouvrage.

Chapitre 2 : Les méthodes scalaires.

Dans ce chapitre sont exposées les méthodes les plus connues de l'optimisation multiobjectif. Nous avons cherché à illustrer ces méthodes à travers des problèmes multiobjectifs analytiques très simples. Quand la résolution de ces exemples ne permet pas d'approfondir la connaissance de la méthode (à cause de calculs trop lourds par exemple), ces exemples n'ont pas été traités en détail.

Chapitre 3 : Les méthodes interactives.

Les méthodes exposées dans ce chapitre sont certainement les plus difficiles à appréhender. Elles sont composées de séquences faisant appel à des méthodes du chapitre 2 et interagissent avec l'utilisateur d'une manière qui n'est pas toujours simple. Ces méthodes étant aujourd'hui beaucoup moins utilisées dans le domaine des sciences de l'ingénieur, la lecture de ce chapitre n'est pas indispensable. Cependant, les lecteurs intéressés par l'aide à la décision pourront lire ce chapitre car, dans ce domaine, les méthodes interactives ont encore une bonne audience.

Chapitre 4 : Les méthodes floues.

La logique floue a un certain succès dans le domaine de l'optimisation multiobjectif. Elle permet de "modéliser un décideur" d'une manière moins contraignante qu'avec les outils traditionnels de l'optimisation multiobjectif et de l'aide à la décision. Les notions requises pour la compréhension de ces méthodes sont présentées en début de chapitre.

Chapitre 5 : Les méthodes exploitant une métaheuristique.

Ce chapitre mériterait une présentation plus détaillée qui occuperait un à plusieurs volumes. Nous avons pris le parti de présenter les éléments indispensables à la compréhension de ce sujet. Nous avons aussi présenté les approches les plus répandues dans le domaine des sciences de l'ingénieur. Le lecteur intéressé trouvera dans la bibliographie une liste de références avec, pour certains articles, des liens permettant leur téléchargement.

Chapitre 6 : Les méthodes d'aide à la décision.

Encore un thème qui nécessiterait un à plusieurs volumes pour une présentation exhaustive. Nous avons choisi encore une fois de présenter les méthodes les plus représentatives de ce domaine. La présentation de ces méthodes s'inspire de quelques ouvrages de référence. Cependant, le lecteur intéressé pourra consulter ces ouvrages pour une présentation détaillée des méthodes.

Partie II : Evaluation des méthodes et critères de choix

Chapitre 7 : La mesure des performances.

Nous avons collecté dans ce chapitre des outils permettant de mesurer les performances des méthodes d'optimisation multiobjectif. Le fonctionnement de ces outils est illustré sur des exemples simples.

Chapitre 8 : Les fonctions de test des méthodes d'optimisation multiobjectif.

Tout comme l'optimisation monobjectif, l'optimisation multiobjectif a besoin de problèmes permettant d'évaluer l'efficacité des méthodes. Nous présentons d'abord la famille des problèmes tests de Deb. Ceux-ci, très répandus, permettent de régler de nombreux paramètres de l'espace de recherche et illustrent très bien les difficultés spécifiques que l'on est susceptible de rencontrer en optimisation multiobjectif. Nous présentons aussi une famille de problèmes tests contraints qui est, à l'heure actuelle, la seule famille cohérente de problèmes tests contraints.

Chapitre 9 : Tentatives de classification des méthodes d'optimisation multiobjectif.

La diversité des méthodes est un atout. Cependant, elle soulève la question suivante : comment choisir une méthode adaptée à la résolution d'un problème donné ?

Ce chapitre propose quelques techniques permettant de lever “partiellement” cette difficulté.

Partie III : Etudes de cas

Chapitre 10 : Etude de cas n°1 : qualification de code scientifique par optimisation multiobjectif.

Nous commençons ici une présentation d'applications illustrant l'utilisation concrète des méthodes d'optimisation multiobjectif dans divers domaines.

Ce chapitre traite de la “qualification” des codes scientifiques, c'est-à-dire de la manière de régler les paramètres d'un code scientifique pour que les résultats fournis par celui-ci soient le plus proches possible de la réalité. Le problème est ici résolu dans l'espace des paramètres et non dans l'espace des fonctions objectif comme on le fait classiquement.

Ce chapitre a été rédigé en collaboration avec M. Dumas du CEA.

Chapitre 11 : Etude de cas n°2 : étude multicritère de l'extension d'un réseau de télécommunications.

L'application illustrée dans ce chapitre concerne le dimensionnement d'un réseau de télécommunications. C'est un champ d'application majeur de l'optimisation multiobjectif et particulièrement d'actualité.

Ce chapitre a été rédigé par C. Decouchon-Brochet, France Télécom - Recherche et Développement.

Chapitre 12 : Etude de cas n°3 : outils décisionnels multicritères utilisés par EADS Launch Vehicles pour traiter les appels d'offres.

Dans ce chapitre, une application d'une méthode d'aide à la décision (la méthode ELECTRE TRI) est présentée. Cette application concerne le classement d'appels d'offres en différentes catégories :
- les appels d'offres que l'on est sûr de gagner,
- les appels d'offres dont l'issue est incertaine,
- les appels d'offres que l'on est sûr de perdre.

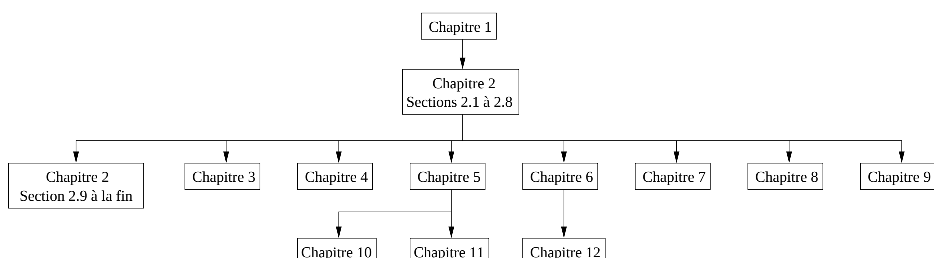
Ce chapitre a été rédigé par J.-M. Contant, J.-L. Bussière et G. Piffaretti de la société EADS Launch Vehicles.

Conclusions : En conclusion, nous récapitulons les atouts et les difficultés de l'optimisation multiobjectif, avant de passer en revue quelques perspectives d'évolution du domaine.

Plan de lecture

Plusieurs lectures sont possibles :

– La lecture linéaire de l'ouvrage.
  Nous conseillons ce mode de lecture aux lecteurs désirant découvrir le domaine de l'optimisation multiobjectif.
– La lecture spécifique de certaines parties de l'ouvrage.
  Si le lecteur est plutôt intéressé par certains domaines particuliers comme l'optimisation interactive, ou les méthodes d'optimisation multiobjectif utilisant des métaheuristiques, il est possible de faire une lecture "au plus court" de cet ouvrage. Certains chapitres sont toutefois nécessaires à la bonne compréhension d'autres chapitres. Nous avons représenté les plus courts chemins de lecture possibles sur la figure suivante :



Remerciements

Les auteurs tiennent à remercier Electricité de France, Recherche et Développement, qui a accueilli Yann Collette pour la préparation de son doctorat, et qui a autorisé la publication de cet ouvrage. Ils adressent également leurs remerciements à toutes les personnes qui ont permis d'illustrer l'ouvrage par la présentation de trois applications industrielles : en particulier, M. Dumas, du CEA, pour sa contribution à la rédaction du chapitre 10 ; C. Decouchon-Brochet, de France Télécom, pour sa rédaction complète du chapitre 11 ; J.-M. Contant, J.-L. Bussière et G. Piffaretti, d'EADS Launch Vehicles, pour leur rédaction complète du chapitre 12.

Yann Collette tient également à remercier Isabelle et Marie Collette qui l'ont soutenu constamment, et ont accepté avec le sourire les longues heures consacrées à l'ouvrage. Il souhaite aussi remercier ses collègues de travail de l'équipe SINETICS/I29 d'EdF R&D, à Clamart, pour leur constante bonne humeur, ainsi que pour leurs commentaires toujours pertinents.

Première partie

Principe des méthodes d'optimisation multiobjectif

Chapitre 1

Introduction : optimisation multiobjectif et dominance

1.1 Qu'est-ce qu'un problème d'optimisation ?

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (de l'optimum donc) d'une fonction donnée. On peut aussi trouver des problèmes d'optimisation pour lesquels les variables de la fonction à optimiser sont contraintes d'évoluer dans une certaine partie de l'espace de recherche. Dans ce cas, on a une forme particulière de ce que l'on appelle un problème d'optimisation sous contraintes.

Ce besoin d'optimisation vient de la nécessité de l'ingénieur de fournir à l'utilisateur un système qui réponde au mieux au cahier des charges. Ce système devra être calibré de manière à :

– occuper le volume minimum nécessaire à son bon fonctionnement (coût des matières premières),
– consommer le minimum d'énergie (coût de fonctionnement),
– répondre à la demande de l'utilisateur (cahier des charges).

Mathématiquement parlant, un problème d'optimisation se présentera sous la forme suivante :

$$\left|\begin{array}{lll} \text{minimiser} & f(\overrightarrow{x}) & \text{(fonction à optimiser)} \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 & (m \text{ contraintes d'inégalité}) \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 & (p \text{ contraintes d'égalité}) \end{array}\right.$$

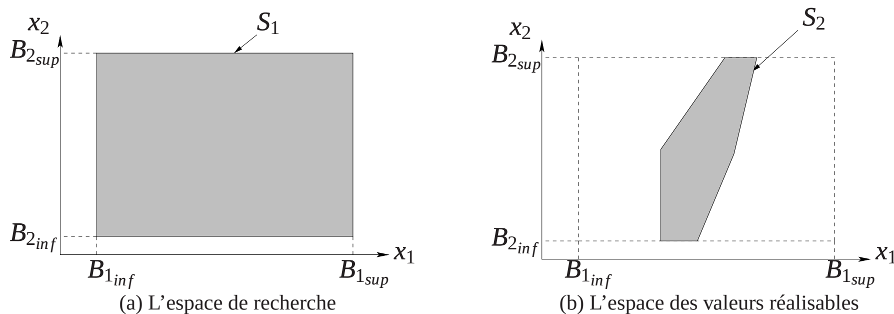
On a $\overrightarrow{x} \in \mathbb{R}^n$, $\overrightarrow{g}(\overrightarrow{x}) \in \mathbb{R}^m$ et $\overrightarrow{h}(\overrightarrow{x}) \in \mathbb{R}^p$.

Ici, les vecteurs $\overrightarrow{g}(\overrightarrow{x})$ et $\overrightarrow{h}(\overrightarrow{x})$ représentent respectivement m contraintes d'inégalité et p contraintes d'égalité.

Cet ensemble de contraintes délimite un espace restreint de recherche de la solution optimale.

En général, on trouve deux types de contraintes d'inégalité :

– Des contraintes du type $B_{i_{inf}} \leq x_i \leq B_{i_{sup}}$: les valeurs de $\overrightarrow{x}$ qui vérifient ces contraintes définissent l' "espace de recherche". Cet espace est représenté à la figure 1.1-a ($n = 2$).
– Des contraintes du type $c(\overrightarrow{x}) \leq 0$ ou $c(\overrightarrow{x}) \geq 0$: les valeurs de $\overrightarrow{x}$ qui vérifient ces contraintes définissent l' "espace des valeurs réalisables". Cet espace est représenté à la figure 1.1-b ($n = 2$).



(a) L'espace de recherche
(b) L'espace des valeurs réalisables

FIG. 1.1 – Les différents espaces de recherche.

1.2 Vocabulaire et définitions

Fonction objectif :
C'est le nom donné à la fonction f (on l'appelle encore **fonction de coût** ou **critère d'optimisation**). C'est cette fonction que l'algorithme d'optimisation va devoir "optimiser" (trouver un optimum).

Sauf mention explicite contraire, on supposera dans la suite que la fonction objectif est à minimiser.

Variables de décision :
Elles sont regroupées dans le vecteur $\overrightarrow{x}$. C'est en faisant varier ce vecteur que l'on recherche un optimum de la fonction f.

Minimum global :
Un "point" $\overrightarrow{x}^*$ est un minimum global de la fonction f si on a :
$f(\overrightarrow{x}^*) < f(\overrightarrow{x})$ quel que soit $\overrightarrow{x}$ tel que $\overrightarrow{x}^* \neq \overrightarrow{x}$. Cette définition correspond au point M_3 de la figure 1.2.

Minimum local fort :
Un "point" $\overrightarrow{x}^*$ est un minimum local fort de la fonction f si on a :
$f(\overrightarrow{x}^*) < f(\overrightarrow{x})$ quel que soit $\overrightarrow{x} \in V(\overrightarrow{x}^*)$ et $\overrightarrow{x}^* \neq \overrightarrow{x}$, où $V(\overrightarrow{x}^*)$ définit un "voisinage" de $\overrightarrow{x}^*$. Cette définition correspond aux points M_2 et M_4 de la figure 1.2.

Minimum local faible :
Un "point" $\overrightarrow{x}^*$ est un minimum local faible de la fonction f si on a :
$f(\overrightarrow{x}^*) \leq f(\overrightarrow{x})$ quel que soit $\overrightarrow{x} \in V(\overrightarrow{x}^*)$ et $\overrightarrow{x}^* \neq \overrightarrow{x}$, où $V(\overrightarrow{x}^*)$ définit un "voisinage" de $\overrightarrow{x}^*$. Cette définition correspond au point M_1 de la figure 1.2.

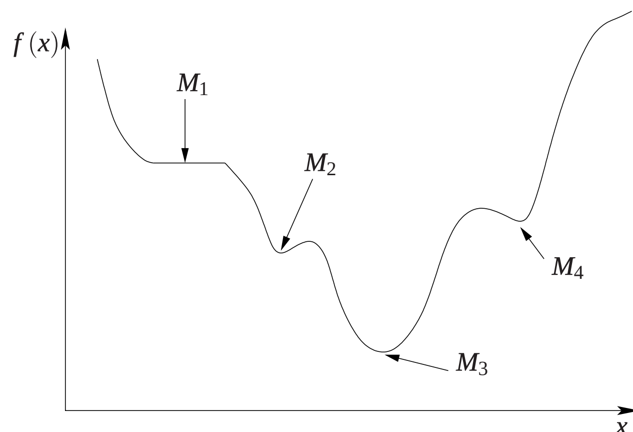


FIG. 1.2 – Les différents minima.

1.3 La classification des problèmes d'optimisation

On peut classer les différents problèmes d'optimisation que l'on rencontre dans la vie courante en fonction de leurs caractéristiques :

1. Nombre de variables de décision :
   – Une ⇒ monovariable.
   – Plusieurs ⇒ multivariable.
2. Type de la variable de décision :
   – Nombre réel continu ⇒ continu.
   – Nombre entier ⇒ entier ou discret.
   – Permutation sur un ensemble fini de nombres ⇒ combinatoire.
3. Type de la fonction objectif :
   – Fonction linéaire des variables de décision ⇒ linéaire.
   – Fonction quadratique des variables de décision ⇒ quadratique.
   – Fonction non linéaire des variables de décision ⇒ non linéaire.
4. Formulation du problème :
   – Avec des contraintes ⇒ contraint.
   – Sans contraintes ⇒ non contraint.

Voici un exemple d'utilisation de cette classification. Si l'on considère le problème de l'optimisation des plans de rechargement de combustible nucléaire, on constate que ce problème est :

– non linéaire ⇒ on n'a pas de représentation explicite du problème. On est obligé d'utiliser un code de calcul numérique pour calculer le plus précisément possible les différentes valeurs à optimiser ;
– contraint ⇒ les éléments de combustible ne peuvent pas être placés n'importe où ;
– multivariable ⇒ un plan de rechargement est composé de plusieurs éléments de combustible dont les caractéristiques sont différentes ;
– combinatoire ⇒ pour passer à un autre plan de rechargement, on permute des éléments de combustible d'un plan de rechargement.

1.4 L'optimisation multiobjectif

La formulation précédente était relative à un problème dans lequel on recherchait un optimum pour une fonction objectif (f dans l'expression précédente).

Cependant, lorsque l'on modélise un problème, on cherche souvent à satisfaire plusieurs objectifs. Par exemple, on veut un système performant et on veut aussi que ce système consomme peu. Dans ce cas, on parle de problème d'optimisation multiobjectif (ou problème d'optimisation multicritère). Celui-ci s'écrit de la manière suivante :

minimiser $\overrightarrow{f}(\overrightarrow{x})$
avec $\overrightarrow{g}(\overrightarrow{x}) \leq 0$
et $\overrightarrow{h}(\overrightarrow{x}) = 0$
où $\overrightarrow{x} \in R^n$, $\overrightarrow{f}(\overrightarrow{x}) \in \mathbb{R}^k$, $\overrightarrow{g}(\overrightarrow{x}) \in \mathbb{R}^m$ et $\overrightarrow{h}(\overrightarrow{x}) \in \mathbb{R}^p$.

On appellera ce problème le **problème P** dans tout le reste de l'ouvrage.

Comme on peut le voir ici, on n'a plus un seul objectif à atteindre, mais k (le vecteur $\overrightarrow{f}$ regroupe k fonctions objectif).

Le but que l'on se fixe dans la résolution d'un problème d'optimisation multiobjectif est de minimiser "au mieux" les différents objectifs. Comme on va le voir dans le paragraphe suivant, dans un problème d'optimisation multicritère, on rencontre souvent des objectifs contradictoires. Deux objectifs sont contradictoires lorsque la diminution d'un objectif entraîne une augmentation de l'autre objectif.

1.5 La multiplicité des solutions

Lorsque l'on cherche à obtenir une solution optimale à un problème d'optimisation multiobjectif donné, on s'attend souvent à trouver une solution et une seule. En fait, on rencontre rarement ce cas de figure. La plupart du temps, on trouve une multitude de solutions, du fait que certains des objectifs sont contradictoires.

En effet, si l'on prend l'exemple du dimensionnement d'une poutre devant supporter une charge donnée, on va vouloir obtenir une poutre de section la plus petite possible, produisant la plus petite déformation possible, lorsque la charge repose sur le milieu de la poutre. Dans cet exemple (représenté à la figure 1.3), de manière intuitive, on s'aperçoit que répondre à l'objectif "poutre de petite section" ne va pas du tout dans le sens de répondre à l'objectif "petite déformation" (voir [Coello et al. 99] pour la modélisation mathématique de ce problème).

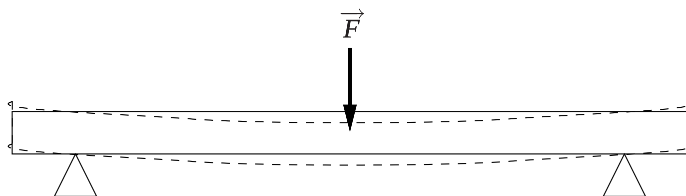


FIG. 1.3 – Déformation d'une poutre subissant une contrainte.

Donc, quand on résoudra un problème d'optimisation multiobjectif, on obtiendra une grande quantité de solutions. Ces solutions, comme on peut s'en douter, ne seront pas **optimales**, au sens où elles ne minimiseront pas tous les objectifs du problème.

Un concept intéressant, qui nous permettra de définir les solutions obtenues, est le **compromis**. En effet, les solutions que l'on obtient lorsque l'on a résolu le problème sont des solutions de compromis. Elles minimisent un certain nombre d'objectifs tout en dégradant les performances sur d'autres objectifs.

1.6 La dominance

1.6.1 Introduction et définitions

Lorsque nous avons résolu notre problème d'optimisation multiobjectif, nous avons obtenu une multitude de solutions. Seul un nombre restreint de ces solutions va nous intéresser. Pour qu'une solution soit intéressante, il faut qu'il existe une relation de **dominance** entre la solution considérée et les autres solutions, dans le sens suivant :

Définition : la relation de dominance
On dit que le vecteur $\overrightarrow{x_1}$ domine le vecteur $\overrightarrow{x_2}$ si :
– $\overrightarrow{x_1}$ est au moins aussi bon que $\overrightarrow{x_2}$ dans tous les objectifs, et,
– $\overrightarrow{x_1}$ est strictement meilleur que $\overrightarrow{x_2}$ dans au moins un objectif.

Les solutions qui dominent les autres mais ne se dominent pas entre elles sont appelées **solutions optimales au sens de Pareto** (ou **solutions non dominées**). On définit comme suit l'optimalité locale et l'optimalité globale au sens de Pareto.

Définition : optimalité locale au sens de Pareto
Un vecteur $\overrightarrow{x} \in \mathbb{R}^n$ est optimal localement au sens de Pareto s'il existe un réel $\Delta > 0$ tel qu'il n'y ait pas de vecteur $\overrightarrow{x}'$ qui domine le vecteur $\overrightarrow{x}$ avec $\overrightarrow{x}' \in \mathbb{R}^n \cap B(\overrightarrow{x}, \Delta)$, où $B(\overrightarrow{x}, \Delta)$ représente une boule de centre $\overrightarrow{x}$ et de rayon Δ.

Un vecteur $\overrightarrow{x}$ est donc optimal localement au sens de Pareto s'il est optimal au sens de Pareto sur une restriction de l'ensemble $\mathbb{R}^n$. Cette définition est illustrée par la figure 1.4.

Définition : optimalité globale au sens de Pareto
Un vecteur $\overrightarrow{x}$ est optimal globalement au sens de Pareto (ou optimal au sens de Pareto) s'il n'existe pas de vecteur $\overrightarrow{x}'$ tel que $\overrightarrow{x}'$ domine le vecteur $\overrightarrow{x}$.

La différence entre cette définition et celle de l'optimalité locale tient dans le fait que l'on ne considère plus une restriction de l'ensemble $\mathbb{R}^n$.

Une version “graphique” de la précédente définition utilise le théorème du contact.

Définition : cône négatif
Un cône négatif est défini dans $\mathbb{R}^k$ de la manière suivante :

$$C^- = \left\{ \overrightarrow{x} \mid \overrightarrow{f}(\overrightarrow{x}) \in \mathbb{R}^k \text{ et } \overrightarrow{f}(\overrightarrow{x}) \leq 0 \right\}$$

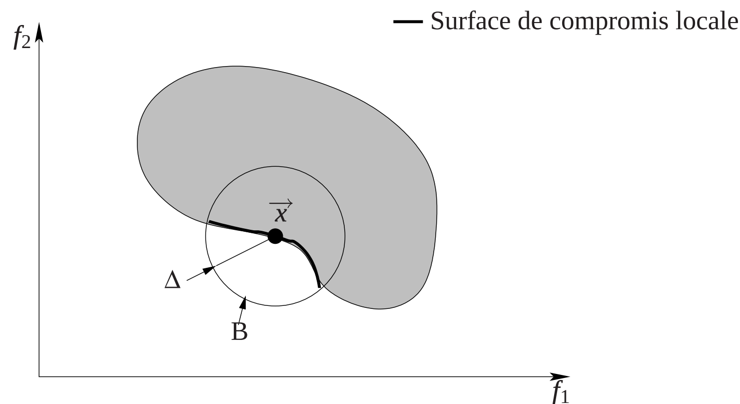


FIG. 1.4 – L'optimalité locale au sens de Pareto.

Définition : le théorème du contact
Un vecteur $\overrightarrow{x}$ est optimal au sens de Pareto pour un problème d'optimisation multiobjectif donné si

$$\left(C^{-}+\overrightarrow{x}\right) \cap F=\{\overrightarrow{x}\}$$

où F désigne l'espace des solutions réalisables.

L'utilisation de ce théorème est illustrée par la figure 1.5.

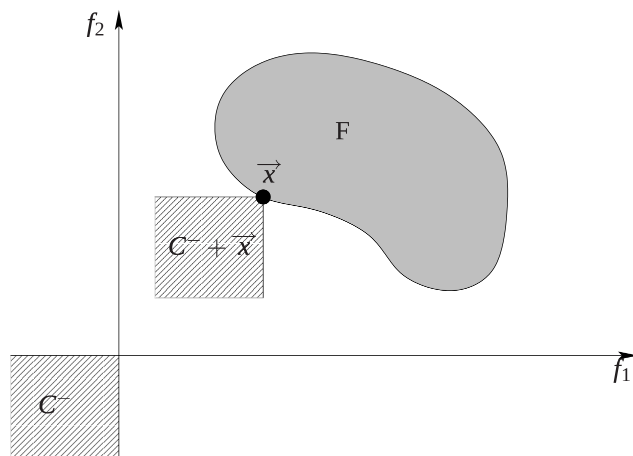


FIG. 1.5 – Le théorème du contact.

Lorsque l'on applique la définition de la dominance, on peut définir quatre régions auxquelles on peut attribuer des niveaux de préférence. Ces régions sont représentées à la figure 1.6. Cette figure reprend le découpage défini par le cône négatif que l'on a introduit précédemment et l'étend à tout l'espace.

Par exemple, si ce graphique est centré sur une solution A et que l'on compare cette solution avec une solution B, on aura les possibilités suivantes :

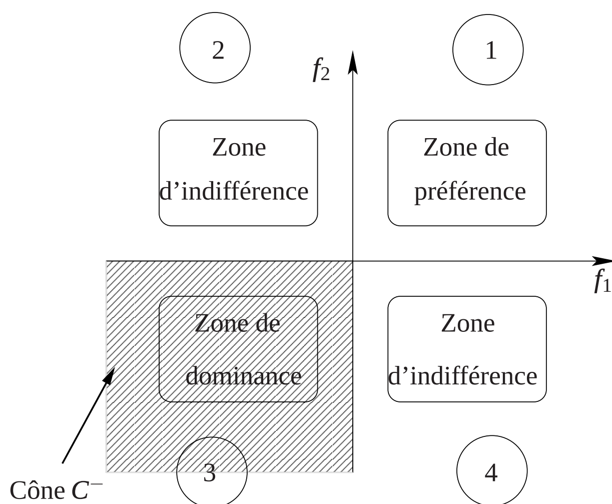


FIG. 1.6 – Les niveaux de préférence dans la relation de dominance.

- si la solution B se trouve dans le quadrant 1, alors la solution A est préférée à la solution B ;
- si la solution B se trouve dans le quadrant 3, alors la solution A est dominée par la solution B ;
- si la solution B se trouve dans l'un des quadrants 2 ou 4, alors, on ne peut pas se prononcer sur la préférence de A par rapport à B ou de B par rapport à A.

1.6.2 Exemple

Pour mieux comprendre ce qu'est une relation de dominance, nous allons traiter un exemple (voir [Deb 99]).

On considère un problème à deux objectifs : maximiser f_1 et minimiser f_2.

Pour ce problème, on trouve un ensemble de solutions. Cet ensemble de solutions est représenté dans le tableau 1.1.

Point	Objectif f_1	Objectif f_2
A	8	5
B	9	2
C	12	1
D	11	2
E	16	2

TAB. 1.1 – Ensemble des solutions d'un problème à deux objectifs.

On représente ces points dans le plan f_1, f_2 (voir figure 1.7).

Nous présentons dans le tableau 1.2 les comparaisons entre les différentes solutions. Précisons d'abord les conventions utilisées :

La comparaison de deux solutions, soit P et Q, se traduit par un couple, à l'intersection de la ligne P et de la colonne Q. Ce couple est constitué de deux symboles, respectivement associés aux objectifs f_1 et f_2 ; chacun de ces symboles peut prendre trois valeurs : +, - ou =,

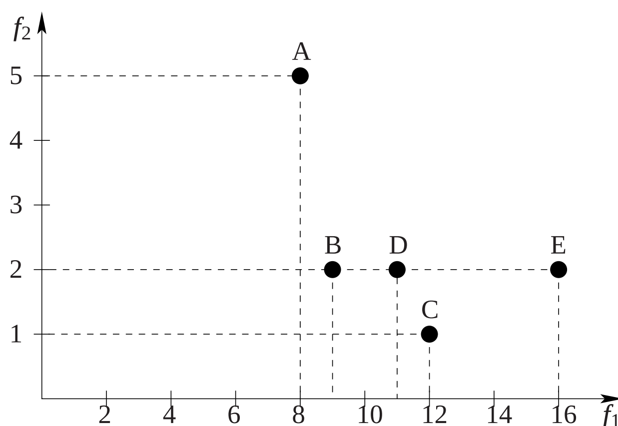


FIG. 1.7 – Représentation des solutions dans le plan f_1, f_2.

selon que P est meilleur, moins bon, ou du même niveau que Q vis-à-vis de l'objectif auquel ce symbole est associé.

On rappelle que l'on cherche à maximiser f_1 et minimiser f_2. Donc, un point P est meilleur qu'un point Q du point de vue de l'objectif f_1 si $f_1(P)$ est plus grand que $f_1(Q)$; un point P est meilleur qu'un point Q du point de vue de l'objectif f_2 si $f_2(P)$ est plus petit que $f_2(Q)$.

Voici un exemple de traitement sur les points A et B du tableau 1.1. Effectuons les comparaisons :

– A est moins bon que B vis-à-vis de l'objectif f_1. Donc, on attribue le signe - au premier élément du couple de la case $[A, B]$.
– A est moins bon que B vis-à-vis de l'objectif f_2. Donc, on attribue le signe - au second élément du couple de la case $[A, B]$.

Si l'on se réfère à la définition précédente, on conclut que la solution B domine la solution A.

	A	B	C	D	E
A		(-,-)	(-,-)	(-,-)	(-,-)
B	(+,+)		(-,-)	(-,=)	(-,=)
C	(+,+)	(+,+)		(+,+)	(-,+)
D	(+,+)	(+,=)	(-,-)		(-,=)
E	(+,+)	(+,=)	(+,-)	(+,=)	

TAB. 1.2 – Classement des solutions.

	A	B	D
A		(-,-)	(-,-)
B	(+,+)		(-,=)
D	(+,+)	(+,=)	

TAB. 1.3 – Classement des solutions de rang 2.

Remarque : D'évidence, les couples situés dans des cases du tableau 1.2 symétriques par rapport à la diagonale principale sont “complémentaires”.

Nous allons maintenant chercher à extraire les solutions non dominées.

1. Considérons le point A,
   – ce point est dominé par les points suivants :
     B(couple (-,-) à l'intersection de la ligne A et de la colonne B),

C (couple (-,-) à l'intersection de la ligne *A* et de la colonne *C*),
D (couple (-,-) à l'intersection de la ligne *A* et de la colonne *D*) et
E (couple (-,-) à l'intersection de la ligne *A* et de la colonne *E*).

2. Considérons le point *B*,
   - ce point est dominé par les points suivants :
     C (couple (-,-) à l'intersection de la ligne *B* et de la colonne *C*),
     D (couple (-,=) à l'intersection de la ligne *B* et de la colonne *D*) et
     E (couple (-,=) à l'intersection de la ligne *B* et de la colonne *E*),
   - il domine le point *A* (couple (+,+) à l'intersection de la ligne *B* et de la colonne *A*).

   On peut dire que le point *A* ne fera pas partie des solutions non dominées de rang 1 car il est possible de trouver un point (*B* en l'occurrence) qui est meilleur que le point *A* sur tous les objectifs.

3. Considérons le point *C*,
   - ce point n'est pas dominé par le point *E* (couple (+,-) à l'intersection de la ligne *E* et de la colonne *C*),
     et il ne domine pas, non plus, le point *E* (couple (-,+) à l'intersection de la ligne *C* et de la colonne *E*) :
     les points *C* et *E* sont donc non dominés ;
   - il domine les points suivants :
     A (couple (+,+) à l'intersection de la ligne *C* et de la colonne *A*),
     B (couple (+,+) à l'intersection de la ligne *C* et de la colonne *B*) et
     D (couple (+,+) à l'intersection de la ligne *C* et de la colonne *D*).

   On peut dire que les points *A*, *B* et *D* ne feront pas partie des solutions non dominées de rang 1 car il est possible de trouver un point (*C* en l'occurrence) qui soit meilleur que ces deux points sur tous les objectifs.

4. Le point *D* étant dominé, il n'est pas nécessaire de le comparer aux autres points.

5. Considérons le point *E*,
   - ce point n'est pas dominé par le point *C* (couple (-,+) à l'intersection de la ligne *C* et de la colonne *E*),
     et il ne domine pas non plus le point *C* (couple (+,-) à l'intersection de la ligne *E* et de la colonne *C*),
   - il domine les points suivants :
     A (couple (+,+) à l'intersection de la ligne *E* et de la colonne *A*),
     B (couple (+,=) à l'intersection de la ligne *E* et de la colonne *B*) et
     D (couple (+,=) à l'intersection de la ligne *E* et de la colonne *D*).

   On constate que les points *E* et *C* sont non dominés. Ils dominent les points *A*, *B* et *D* mais ne se dominent pas entre eux.

On va, maintenant, sortir ces deux points (*E* et *C*) du tableau : ils forment **l'ensemble des points non dominés**. On peut établir un classement des solutions en fonction du **rang de domination**. Dans notre exemple, on attribue le rang 1 (car on a fini la première comparaison) aux points *E* et *C*, parce qu'ils dominent tous les autres points mais ne se dominent pas entre eux. Ces points sont donc des solutions **optimales au sens de Pareto** de rang 1.

On recommence à appliquer la règle sur les éléments restants du tableau.

Les solutions restantes après suppression des points *E* et *C* sont représentées dans le tableau 1.3.

Ce processus s'arrête lorsque l'ensemble des points à comparer est vide. La figure 1.8 représente les différents points et leur rang.

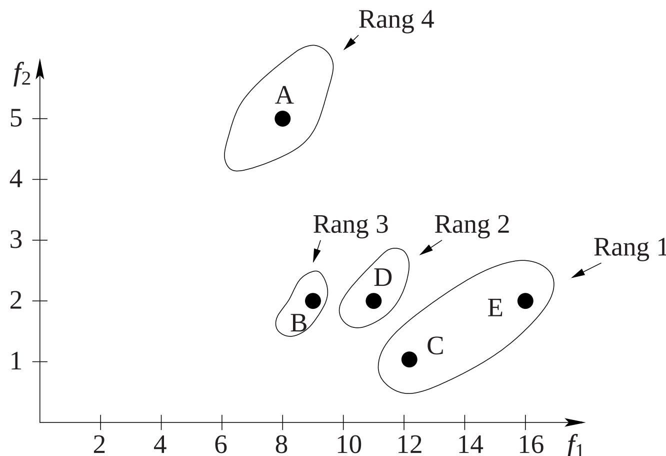


FIG. 1.8 – Les solutions et leurs rangs de Pareto.

Le pseudo-code de la fonction d'assignation du rang de Pareto est représenté dans l'algorithme 1.1. Dans cet algorithme, la variable N désigne le nombre de points de l'ensemble sur lequel on effectue les comparaisons.

Algorithm 1.1 Assignation du rang de Pareto.

```
RangCourant = 1
m = N
while N ≠ 0 do
  For i = 1 to m do
    If X_i est Non dominé
      Rang(X_i, t) = RangCourant
  End For
  For i = 1 to m do
    If Rang(X_i, t) = RangCourant
      Ranger X_i dans une population temporaire
      N = N − 1
  End For
  RangCourant = RangCourant + 1
  m = N
End While
```

1.6.3 Le dimensionnement d'une barre

1.6.3.1 Une barre pleine de section carrée

Nous allons maintenant traiter plus en détail notre exemple d'introduction : le dimensionnement d'une barre de longueur donnée (1 m), représentée à la figure 1.9.

On recherche le côté a de la section carrée, permettant de minimiser le poids de la barre ainsi que sa déformation lorsqu'elle est soumise à une force de 1000 N appliquée en son centre.



FIG. 1.9 – Une barre pleine de section carrée.

Comme minimiser le volume d'une barre de longueur donnée revient à minimiser sa section, nous allons exprimer la section et la déformation maximale en fonction de la longueur du côté a.

$$\begin{cases} S(a) = a^2 \\ d(a) = 1000 + \frac{1 \cdot 10^{-3}}{192 + 2 \cdot 10^5 + \frac{a^4}{12}} \\ a \leq 0.1 \end{cases}$$

(pour les caractéristiques de la barre explicitées dans [Spenle et al. 96])

La seule contrainte que nous nous fixons est que le côté de la section ne dépasse pas 10 cm.

Pour visualiser l'allure de l'ensemble des valeurs réalisables, nous allons choisir un certain nombre de valeurs de a, au hasard, puis nous allons calculer les valeurs des deux objectifs et les tracer dans un plan (S, d). Cette allure est représentée à la figure 1.10.

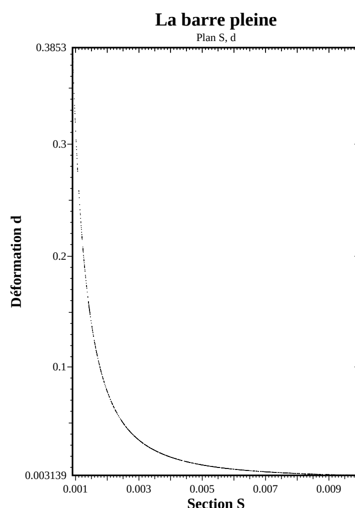


FIG. 1.10 – L'ensemble des valeurs des objectifs pour une barre pleine de section carrée.

La première chose que nous pouvons constater sur cette figure est que les deux objectifs sont antagonistes. La minimisation des valeurs d'un objectif entraîne la maximisation de l'autre objectif. En revanche, sur cette figure, on peut voir que toutes les solutions sont non

dominées. Ceci vient du fait que le problème d'optimisation, tel que nous l'avons écrit, se ramène à une simple courbe paramétrique.

Le second point que nous pouvons souligner est l'intérêt de l'optimisation multiobjectif par rapport à l'optimisation monobjectif. En effet, si nous avions effectué une optimisation monobjectif sur ce problème de dimensionnement, nous aurions cherché soit à optimiser le volume de la barre, soit à optimiser la déformation de cette même barre. Dans les deux cas, nous aurions obtenu une valeur absurde :

- une longueur de côté nulle pour la section de la barre (le volume est ainsi minimisé à outrance) ;
- une longueur de côté démesurée par rapport à la longueur de la barre, dans le cas de l'optimisation de la déformation de la barre. En effet, plus la section est importante, moins la barre se déforme.

Ce problème de dimensionnement montre que, dans certains cas, un problème d'optimisation ne peut pas se ramener à un seul objectif.

1.6.3.2 Une barre creuse de section carrée

Pour mettre en évidence un autre aspect important de l'optimisation multiobjectif, nous allons maintenant légèrement modifier le problème en ajoutant un nouveau paramètre. Nous considérons une barre creuse et de section carrée. La dimension intérieure est désignée par la variable *b*. Les différentes variables représentatives sont indiquées à la figure 1.11.

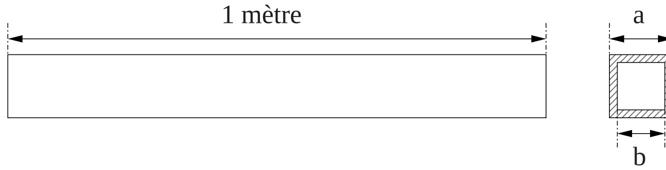


FIG. 1.11 – Une barre creuse de section carrée.

$$\begin{cases} S(a,b) = a^2 - b^2 \\ d(a,b) = 1000 + \frac{1\cdot 10^{-3}}{192 + 2\cdot 10^5 + \frac{a^4 - b^4}{12}} \\ a \leq 0.1 \\ b + 0.04 \leq a \end{cases}$$

Pour visualiser l'allure de l'ensemble des solutions réalisables, nous allons choisir un certain nombre de valeurs du couple (a,b), au hasard pour $a,b \in [0,0.1]$, puis nous allons calculer les valeurs des deux objectifs et les tracer dans un plan (S,d). Cette allure est représentée à la figure 1.12.

Pour cet exemple, on peut maintenant remarquer que la situation est plus compliquée. Les différentes solutions ne forment plus une simple courbe, nous avons désormais un véritable ensemble de solutions. La différence avec l'exemple précédent est que, maintenant, toutes les solutions de cet ensemble ne se valent pas. Un groupe de solutions dominent les autres. Pour extraire ces solutions, nous allons appliquer la relation de dominance et éliminer les points dominés. On obtient alors l'ensemble représenté à la figure 1.13.

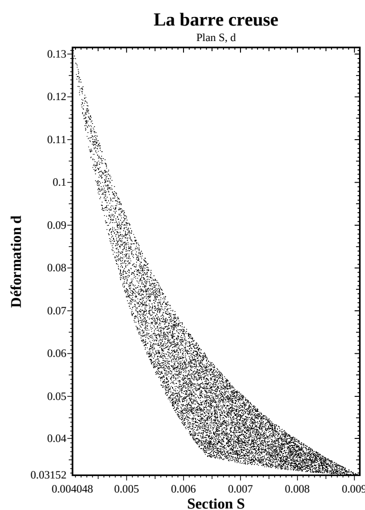


FIG. 1.12 – L'ensemble des valeurs des objectifs pour une barre creuse de section carrée.

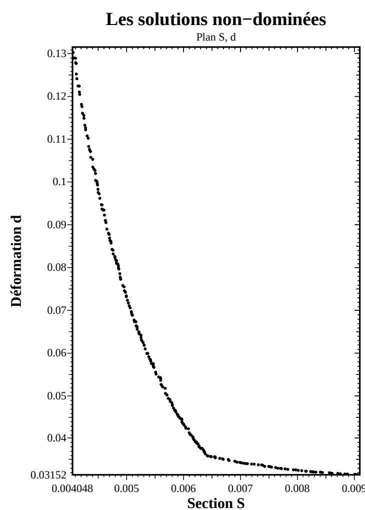


FIG. 1.13 – Les solutions non dominées pour le dimensionnement d'une barre creuse de section carrée.

On peut remarquer que, après filtrage des solutions dominées, seule une frontière de l'ensemble de départ subsiste. Il s'agit de la surface de compromis et c'est sur cette surface de compromis que l'on trouve les "meilleures" solutions, celles que l'on ne pourra pas améliorer sur les deux objectifs.

1.7 Illustration de l'intérêt de l'optimisation multiobjectif

On peut comprendre l'intérêt d'effectuer une optimisation multiobjectif par rapport à une optimisation monobjectif en visualisant l'amélioration relative d'un objectif par rapport à un autre sur un problème test simple.

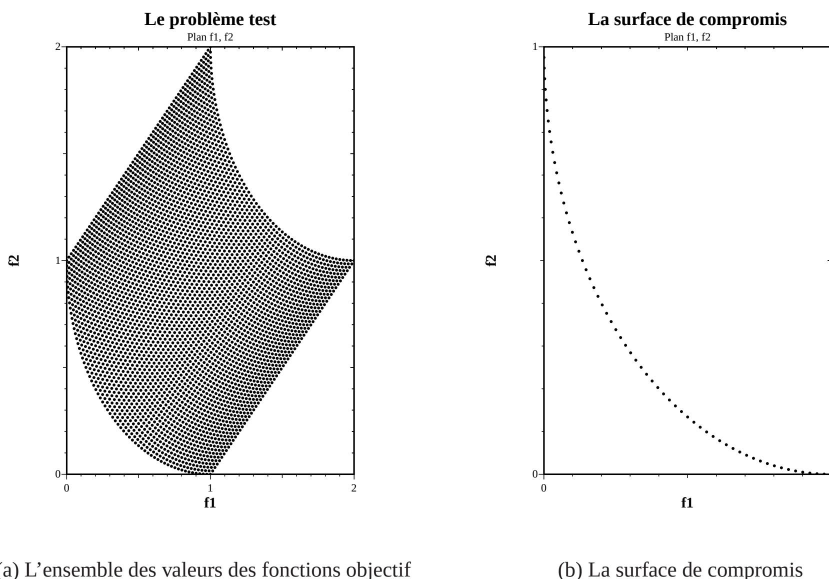
Ce problème est le suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\ ,x) = 1 - \cos(\ ) + x \\ \text{minimiser} & f_2(\ ,x) = 1 - \sin(\ ) + x \\ \text{avec} & 0 \leq \quad \leq \frac{\Pi}{2} \\ \text{et} & 0 \leq x \leq 1 \end{array}\right.$$

La surface de compromis est obtenue en remplaçant la variable x par la valeur 0. On a alors l'expression de la surface de compromis :

$$\left|\begin{array}{l} f_1(\ ) = 1 - \cos(\ ) \\ f_2(\ ) = 1 - \sin(\ ) \\ 0 \leq \quad \leq \frac{\Pi}{2} \end{array}\right.$$

L'ensemble des valeurs des fonctions objectif, ainsi que la surface de compromis, sont représentés à la figure 1.14.

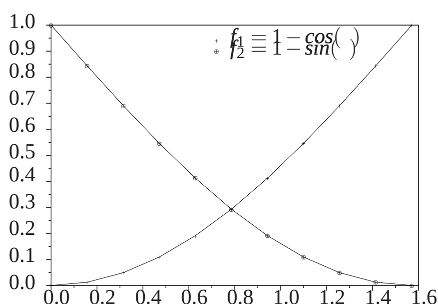


(a) L'ensemble des valeurs des fonctions objectif (b) La surface de compromis

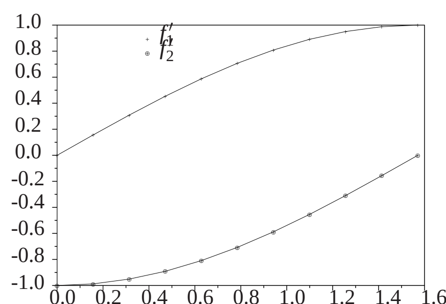
FIG. 1.14 – Le problème test trigonométrique.

Pour cette surface de compromis, nous allons maintenant tracer l'amélioration relative que l'on peut obtenir sur chaque objectif en fonction de sa position sur la surface de compromis. Pour cela, on divise la surface de compromis en dix morceaux (l'angle varie de 0 à 90° par pas de 9°). L'écart relatif est mesuré par rapport à l'excursion totale de la fonction objectif (cette excursion vaut 1 pour les deux fonctions objectif).

	$f_1(\ )$	$f_2(\ )$
0	0	1
9	0.0123	0.843
18	0.0489	0.691
27	0.109	0.546
36	0.191	0.412
45	0.293	0.293
54	0.412	0.191
63	0.546	0.109
72	0.691	0.0489
81	0.843	0.0123
90	1	0



(a) Evolution absolue



(b) Evolution relative

FIG. 1.15 – L'évolution des valeurs des fonctions objectif sur la surface de compromis.

Comme on peut le voir à la figure 1.15-a, suivant la position du point sur la surface de compromis, on peut espérer une bonne réduction du niveau d'une fonction objectif sans avoir d'augmentation très importante du niveau de l'autre fonction objectif. C'est le cas, par exemple, de la fonction objectif f_1 pour des valeurs de   proches de 0. Pour cette position, si l'on augmente la valeur de   de 0.2 radian (on passe ainsi de 0 radian à 0.2 radian), la valeur de la fonction objectif f_1 passe de 1 à 0.843 alors que la fonction objectif f_2 passe elle de 0 à 0.0123.

Le gain que l'on peut espérer en un point précis peut être représenté par la dérivée de la fonction objectif par rapport à la variable   en ce point. Une dérivée positive signifie une amélioration de la fonction objectif alors qu'une dérivée négative signifie une détérioration de la valeur de la fonction objectif.

Ce comportement typique de l'optimisation multiobjectif reflète assez bien le but poursuivi par cette discipline : minimiser un groupe de fonctions objectif sans trop dégrader les valeurs des optima obtenus par rapport à ceux obtenus lors d'une optimisation monobjectif effectuée objectif par objectif.

1.8 Les relations dérivées de la dominance

1.8.1 Introduction

La relation de dominance n'offre pas de degrés de liberté dans sa définition. Par exemple, il n'est pas possible d'inclure dans la définition de la relation de dominance une préférence d'un objectif par rapport à un autre. C'est pour contrecarrer ce manque de flexibilité que des relations dérivées de la relation de dominance ont été développées. Les solutions que permettent de trouver ces relations dérivées de la dominance sont toutes optimales au sens de Pareto. La grande différence que l'on rencontre avec ces relations est que l'ensemble des solutions que l'on obtient avec ces relations est un sous-ensemble de l'ensemble des solutions obtenues avec la relation de dominance de Pareto.

Dans cette section, l'ensemble S^k désigne l'ensemble des solutions réalisables d'un problème d'optimisation à k fonctions objectif.

1.8.2 Optimalité lexicographique

Cette définition de l'optimalité permet d'inclure une préférence entre objectifs [Ehrgott 97].

Définition : optimalité au sens lexicographique
Une solution $\overrightarrow{x^*} \in S^k$ est optimale au sens lexicographique si
$\overrightarrow{x^*} \leq_{lex} \overrightarrow{x}, \forall \overrightarrow{x} \in S^k - \left\{\overrightarrow{x^*}\right\}$.

Si $\overrightarrow{x}, \overrightarrow{y} \in S^k$, on dit que $\overrightarrow{x} \leq_{lex} \overrightarrow{y}$ s'il existe une valeur d'index q^* telle que $x_q = y_q$ pour $q = 1, \cdots, (q^* - 1)$ et $x_{q^*} < y_{q^*}$. Les relations entre x_q et y_q pour $q \geq q^*$ ne sont pas prises en compte car nous nous arrêtons à l'indice q^* (c'est le premier indice pour lequel $x_q < y_q$).

Cette définition implique que l'utilisateur ait rangé par ordre d'importance les différents objectifs. La comparaison entre les deux solutions se fera dans l'ordre de classement des objectifs.

Illustrons l'utilisation de cette relation en prenant un exemple.

Soient deux points A et B :

$$\begin{array}{rcl} A & = & (1,2,3,4,5,6) \\ B & = & (1,2,3,9,4,9) \end{array}$$

Pour ces deux points, on a $A \leq_{lex} B$ car, jusqu'à la troisième position, on a $A_i = B_i$, $i = 1,2,3$ et, pour la quatrième position, on a $4 < 9$. On conclut donc que la solution A domine lexicographiquement la solution B.

1.8.3 Optimalité extrême

Comme pour la relation d'optimalité lexicographique, cette relation permet d'établir une préférence entre critères. Cette préférence est établie en utilisant des poids. Plus un objectif sera important, plus son poids sera élevé [Ehrgott 97].

Définition : optimalité extrême
Une solution $\overrightarrow{x^*} \in S^k$ est extrême-optimale si, étant donné un vecteur de poids $\overrightarrow{\ } \in \mathbb{R}^k$ tel que $\oplus_i \ \ _i = 1$, $\overrightarrow{x^*}$ est une solution optimale du problème de minimisation monocritère ayant pour fonction objectif

$$\bigoplus_{i=1}^{k} \ _i \cdot \overrightarrow{x_i} \tag{1.1}$$

donc :

$$\bigoplus_{i=1}^{k} \ _i \cdot \overrightarrow{x_i^*} \leq \bigoplus_{i=1}^{k} \ _i \cdot \overrightarrow{x_i}, \forall \overrightarrow{x} \in S^k - \left\{\overrightarrow{x^*}\right\} \tag{1.2}$$

Illustrons la notion d'extrême-dominance en reprenant l'exemple précédent. Ici, on considère l'objectif 6 comme objectif de référence. On considère aussi que les objectifs $1,3$ et 5 sont 20 % plus importants que l'objectif de référence et que les objectifs 2 et 4 sont équivalents à l'objectif de référence. On calcule maintenant les poids de chaque objectif :

$$\begin{array}{l} w_1 = w_3 = w_5 = 1.2 \cdot w_6 \\ w_2 = w_4 = w_6 \\ \oplus_{i=1}^{6} w_i = 1 \end{array}$$

La résolution de ces équations nous donne :

$$\begin{array}{l} w_1 = w_3 = w_5 = \frac{0.2}{1.1} = 0.18 \\ w_2 = w_4 = w_6 = \frac{1}{6.6} = 0.15 \end{array}$$

$\oplus_{i=1}^{6} w_i \cdot A_i = 3.45$ et $\oplus_{i=1}^{6} w_i \cdot B_i = 5.39$ donc, le point A extrême-domine le point B.

1.8.4 Optimalité maximale

Cette relation, contrairement aux précédentes, ne permet pas d'introduire une préférence entre objectifs [Ehrgott 97].

Définition : optimalité maximale
Une solution $\overrightarrow{x^*} \in S^k$ est max-optimale si la valeur du pire objectif est aussi petite que possible :

$$\max_{q \in \{1,\cdots,k\}} x_q^* \leq \max_{\substack{q \in \{1,\cdots,k\} \\ \overrightarrow{x} \in S^k - \left\{\overrightarrow{x^*}\right\}}} x_q \tag{1.3}$$

Illustrons cette relation en prenant l'exemple précédent. On peut dire que la solution A max-domine la solution B car :

$$\max A = 6 < \max B = 9$$

1.8.5 La cône optimalité

Une autre relation de classement existe. Il s'agit de la relation de cône-dominance. Cette relation possède l'avantage, par rapport à la relation de Pareto, d'être "réglable".

Définition : un cône
Un cône de pente   est défini de la manière suivante :
– Si $0 < \quad < 1$ (voir figure 1.16-a) :

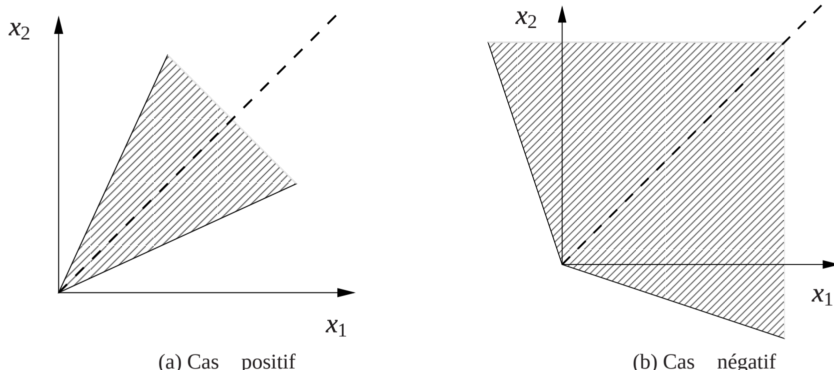
$$C\ (x_1, x_2) = \left\{ \overrightarrow{x} \in \mathbb{R}^2 \,|\, x_1 \geq 0, x_2 \geq 0, \quad \cdot x_1 \leq x_2 \leq \frac{1}{\ } \cdot x_1 \right\} \tag{1.4}$$

– Si $\quad < 0$ (voir figure 1.16-b) :

$$C\ (x_1, x_2) = \left\{ \overrightarrow{x} \in \mathbb{R}^2 \,|\quad \cdot x_1 \leq x_2 \text{ et } \quad \cdot x_2 \leq x_1 \right\} \tag{1.5}$$

– Si $\quad = 0$:

$$C\ (x_1, x_2) = \left\{ \overrightarrow{x} \in \mathbb{R}^2 \,|\, x_1 \geq 0 \text{ et } x_2 \geq 0 \right\}$$



(a) Cas   positif (b) Cas   négatif

FIG. 1.16 – L'allure du cône en fonction de .

Cette relation peut être étendue au cas multidimensionnel. On trouvera dans [Deb 01] une relation différente permettant aussi de régler la forme du cône de domination.

Définition : le cône multidimensionnel
Un cône de pente   est défini de la manière suivante :

$$\begin{aligned} C\ = \ & \left\{ \overrightarrow{x} \in \mathbb{R}^n \,|\, \forall i \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, n\}, j > i, \right. \\ & \left. x_i, x_j \in C\ (x_i, x_j) \right\} \end{aligned} \tag{1.6}$$

La projection du cône multidimensionnel sur des plans est représentée à la figure 1.17.

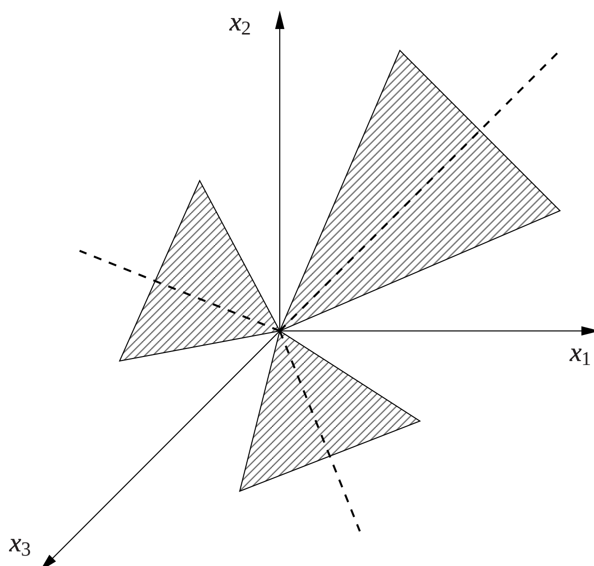
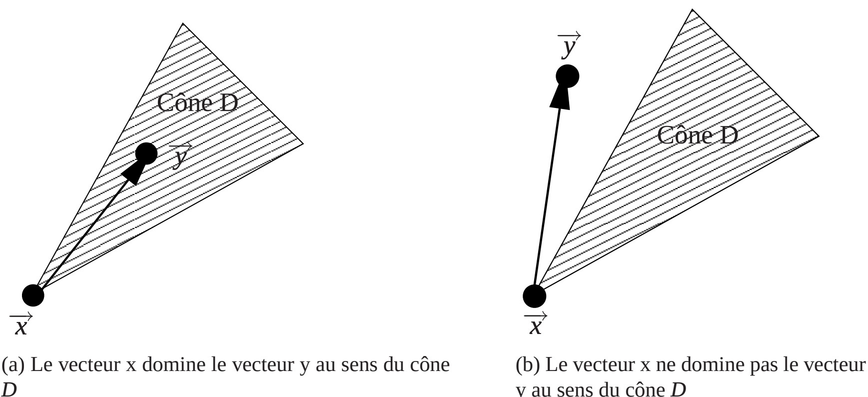


FIG. 1.17 – La projection du cône multidimensionnel.

Définition : la cône-dominance
Soit un cône C . Un vecteur $\overrightarrow{x} \in \mathbb{R}^n$ cône-domine un vecteur $\overrightarrow{y} \in \mathbb{R}^n$ (cette relation est notée $\overrightarrow{x} \leq_C \overrightarrow{y}$) si $\overrightarrow{y} - \overrightarrow{x} \in C$ et $\overrightarrow{x} \neq \overrightarrow{y}$ (ou encore $\overrightarrow{y} - \overrightarrow{x} \in C - \{0\}$).

La relation de cône-dominance est représentée à la figure 1.19. Nous avons aussi représenté deux exemples d'application de cette relation à la figure 1.18.

Sur ces deux figures, nous avons représenté le "cône de domination". Il s'agit de la zone dans laquelle le point d'origine (le vecteur $\overrightarrow{x^*}$ dans la définition de la cône-dominance) domine tous les points appartenant à cette zone (les vecteurs $\overrightarrow{x}$ dans la définition de la cône-dominance).



(a) Le vecteur x domine le vecteur y au sens du cône D

(b) Le vecteur x ne domine pas le vecteur y au sens du cône D

FIG. 1.18 – Deux exemples d'application de la relation de cône-dominance.

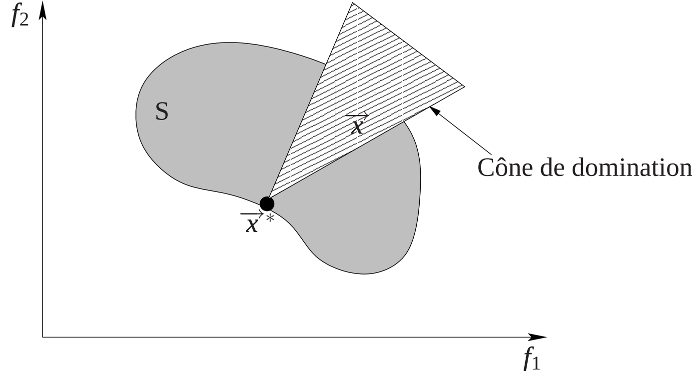


FIG. 1.19 – La relation de cône-dominance.

La relation de cône-dominance est équivalente à la relation de dominance que nous avons vue dans la section précédente lorsque $= 0$.

1.8.6 La a-dominance

Cette définition de dominance a été introduite dans [Othmani 98].

Définition : a-dominance
Une solution $\overrightarrow{x^*} \in S^k$ a-domine une solution $\overrightarrow{x} \in S^k$ s'il existe un ensemble de combinaisons de $k+1-a$ critères (on note $I_{(k+1-a)}$ l'ensemble des index correspondant à l'ensemble des combinaisons de ces critères) tel que :

1. $\overrightarrow{x_j^*} \leq \overrightarrow{x_j}$ pour tout $j \in I_{(k+1-a)}$, et
2. $\overrightarrow{x_j^*} < \overrightarrow{x_j}$ pour au moins un $j \in I_{(k+1-a)}$.

Pour illustrer cette définition, prenons un exemple. Considérons une famille de trois critères (C_1, C_2 et C_3), deux solutions $\overrightarrow{x_A}$ et $\overrightarrow{x_B}$ et cherchons à établir la 2-dominance.

Pour la 2-dominance, on doit tester la relation de dominance entre les vecteurs $\overrightarrow{x_A}$ et $\overrightarrow{x_B}$ sur toutes les combinaisons de $3+1-2=2$ critères. Ces familles sont les suivantes :
– $F_1 = \{C_1, C_2\} \Rightarrow I_1^{'} = \{1,2\}$
– $F_2 = \{C_1, C_3\} \Rightarrow I_2^{'} = \{1,3\}$
– $F_3 = \{C_2, C_3\} \Rightarrow I_3^{'} = \{2,3\}$

Donc, $I_2 = \left\{I_1^{'}, I_2^{'}, I_3^{'}\right\} = \{\{1,2\},\{1,3\},\{2,3\}\}$.

Considérons maintenant deux points A et B.

	C_1	C_2	C_3
A	1	2	3
B	1	1	2

Le point A 2-domine le point B car il domine le point B sur chaque famille de critères :
– A domine B si l'on considère les critères 1 et 2,
– A domine B si l'on considère les critères 1 et 3,

– A domine B si l'on considère les critères 2 et 3.

Donc, A 2-domine B car A domine B si l'on considère les combinaisons de critères $\{C_1, C_2\}$, $\{C_1, C_3\}$ et $\{C_2, C_3\}$.

1.8.7 La dominance au sens de Geoffrion

Une dernière forme de dominance importante dans le monde de l'optimisation multiobjectif est la dominance au sens de Geoffrion (voir [Ehrgott 00] et [Miettinen 99]). Les solutions optimales obtenues par ce type de dominance sont appelées les solutions Pareto optimales propres.

Définition : la dominance au sens de Geoffrion
Une solution $x^* \in S$ est appelée solution Pareto optimale propre si :
- elle est Pareto optimale,
- il existe un nombre $M > 0$ tel que $\forall i$ et $\forall x \in S$ vérifiant $f_i(x) < f_i(x^*)$, il existe un index j tel que $f_j(x^*) < f_j(x)$ et :

$$\frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \leq M$$

Cette relation n'est quasiment jamais utilisée telle quelle. En général, on utilise plutôt un résultat qui découle de cette définition. En effet, l'interprétation de ce théorème faite dans [Ehrgott 00] est la suivante :

> "Les solutions Pareto optimales propres ont des compromis bornés suivant leurs objectifs."

Un théorème relatif à la méthode de pondération des fonctions objectif utilisant ce résultat est le suivant :

Théorème :
Soit la méthode d'agrégation des fonctions objectif suivantes :

$$f_{eq}(\overrightarrow{x}) = \bigoplus_{i=1}^{N} \ {}_i \cdot f_i(\overrightarrow{x})$$

Supposons $\forall i = 1, \cdots, N, \quad {}_i > 0$ avec $\oplus_{i=1}^{N} \ {}_i = 1$.
Si x^* est une solution optimale obtenue en utilisant la méthode d'agrégation ci-dessus, alors cette solution est aussi Pareto optimale propre.

La méthode de pondération des fonctions objectif, avec des poids qui respectent les relations ci-dessus, permet d'obtenir des solutions avec des compromis bornés.

1.8.8 Conclusion

Ces différents types de relations de dominance permettent d'avoir suffisamment de degrés de liberté pour choisir une relation qui reproduise au mieux le comportement d'un ingénieur ou d'un décideur. Par exemple, la relation de dominance lexicographique permet de reproduire le comportement d'un décideur qui choisirait une solution parmi N solutions en considérant des critères rangés par ordre d'importance. En effet, celui-ci, quand il compare deux

solutions, compare les valeurs des objectifs, jusqu'à ce qu'un objectif permette de déterminer une préférence entre les deux solutions considérées. Une fois que la préférence entre les deux solutions est déterminée, on ne considère plus les objectifs restants. La comparaison s'arrête là.

La dominance est donc un outil, parmi d'autres, permettant de reproduire une démarche de recherche d'optimum. Elle n'est pas la seule, bien sûr, mais elle est un élément important pour la résolution d'un problème.

1.9 La surface de compromis

Le petit nombre de solutions de rang 1 que l'on a sélectionnées en utilisant la règle de classement basée sur la définition de la dominance forme ce que l'on appelle la **surface de compromis** (ou **front de Pareto**).

Imaginons que nous ayons un problème à deux objectifs (minimiser f_1 et minimiser f_2 sous les contraintes $\overrightarrow{g}(\overrightarrow{x}) \leq 0$ et $\overrightarrow{h}(\overrightarrow{x}) = 0$) :

- On appelle S l'ensemble des valeurs du couple $(f_1(\overrightarrow{x}), f_2(\overrightarrow{x}))$ quand $\overrightarrow{x}$ respecte les contraintes $\overrightarrow{g}(\overrightarrow{x})$ et $\overrightarrow{h}(\overrightarrow{x})$.
- On appelle P la surface de compromis.

On représente S et P sur la figure 1.20.

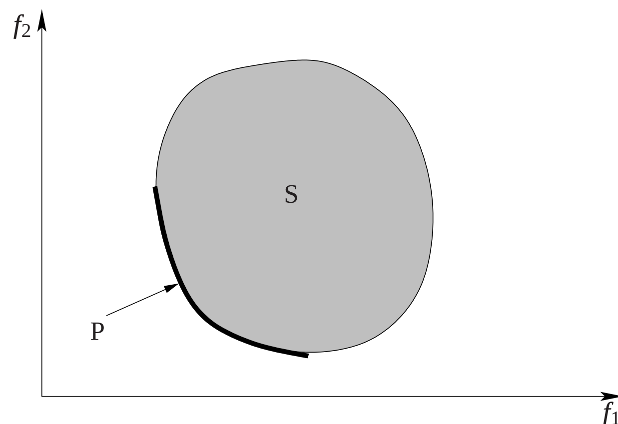
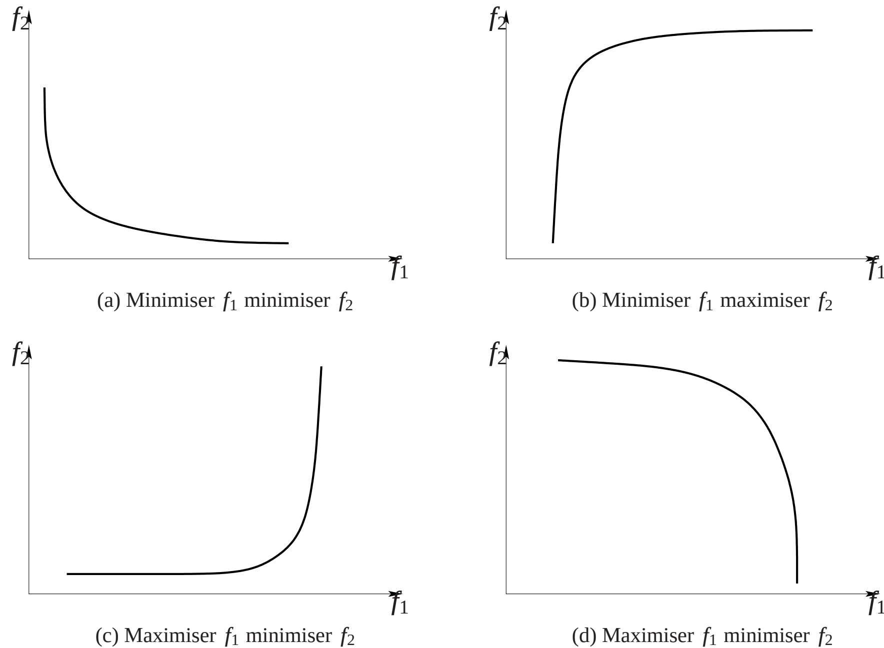


FIG. 1.20 – Représentation de la surface de compromis.

Une propriété est remarquable : en fonction du type de problème que l'on cherche à résoudre, on obtient une forme de surface de compromis. Les formes les plus courantes de surfaces de compromis sont réunies à la figure 1.21. Ces formes de surfaces de compromis sont typiques d'un problème d'optimisation multiobjectif sur un ensemble de solutions convexe (pour une définition de la convexité, voir section 1.10). C'est ce type d'ensemble que l'on rencontre la plupart du temps.

On observe deux points caractéristiques associés à une surface de compromis :

Point idéal :
Les coordonnées de ce point sont obtenues en optimisant chaque fonction objectif séparément.



(a) Minimiser f_1 minimiser f_2

(b) Minimiser f_1 maximiser f_2

(c) Maximiser f_1 minimiser f_2

(d) Maximiser f_1 minimiser f_2

FIG. 1.21 – Formes les plus courantes de surfaces de compromis dans le cas de deux objectifs.

Point ”nadir” :
Les coordonnées de ce point correspondent aux pires valeurs obtenues par chaque fonction objectif lorsque l'on restreint l'espace des solutions à la surface de compromis.

Le point idéal est utilisé dans beaucoup de méthodes d'optimisation comme point de référence. Le point nadir, lui, sert à restreindre l'espace de recherche ; il est utilisé dans certaines méthodes d'optimisation interactives.

Ces deux définitions sont illustrées à la figure 1.22.

1.10 La convexité

Comme nous le verrons plus loin, certaines méthodes d'optimisation multiobjectif nécessitent de respecter certaines hypothèses. Le plus souvent, la méthode réclame de travailler sur un espace S des valeurs de $\overrightarrow{f}$ qui soit convexe.

Définition : la convexité
Un ensemble S est convexe si, étant donnés deux points distincts quelconques de cet ensemble, le segment qui relie ces deux points est contenu dans l'ensemble S.

Un exemple d'ensemble convexe et un exemple d'ensemble non convexe sont représentés à la figure 1.23.

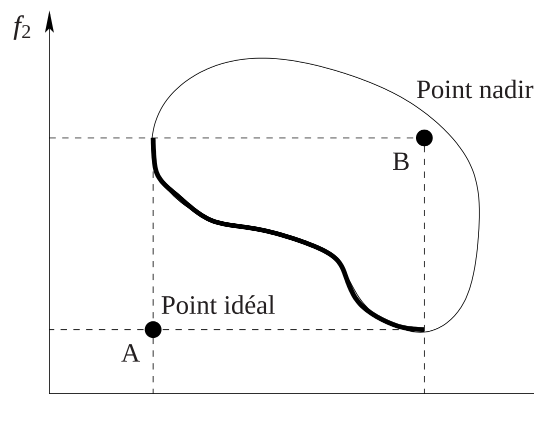


FIG. 1.22 – Représentation du point idéal et du point “nadir”.

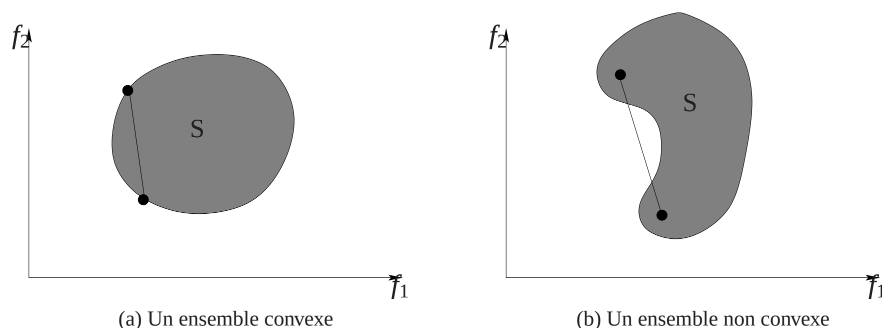


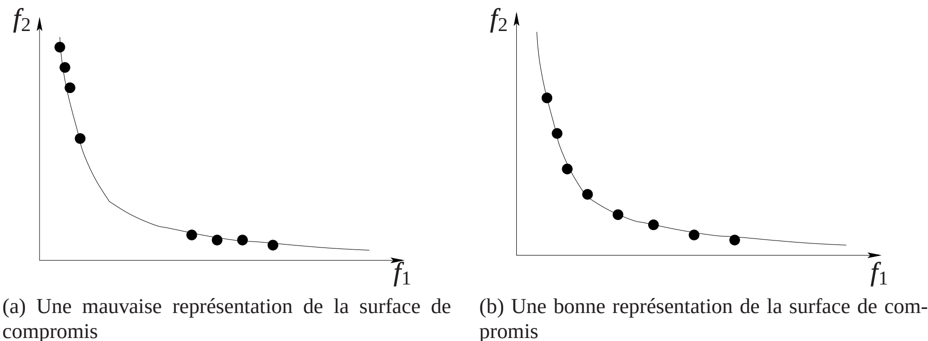
FIG. 1.23 – Exemples d’ensemble convexe et d’ensemble non convexe.

1.11 La représentation de la surface de compromis

Toutes les représentations de la surface de compromis, pour un même problème, ne sont pas équivalentes. En effet, la représentation idéale de la surface de compromis devra être constituée de points solution de notre problème répartis de manière uniforme sur la surface de compromis (voir figure 1.24).

Dans le premier cas, les points représentant la surface de compromis ne sont pas répartis de manière uniforme. L’utilisateur n’aura alors pas en sa possession un ensemble de solutions très utile. En effet, s’il décide que la solution qu’il avait choisie ne lui convient pas, le choix d’une autre solution risque de faire varier brusquement tous ses objectifs, et cette nouvelle solution ne lui conviendra pas non plus. Il est alors probable que la solution offrant le “meilleur” compromis se trouve dans une zone qui ne soit pas représentée par des points solution.

La détermination d’une bonne représentation de la surface de compromis sera un critère de choix d’une méthode d’optimisation multiobjectif.



(a) Une mauvaise représentation de la surface de compromis

(b) Une bonne représentation de la surface de compromis

FIG. 1.24 – La représentation de la surface de compromis.

1.12 Les méthodes de résolution des problèmes d'optimisation multiobjectif

Il existe un nombre important de méthodes et nous avons classé celles-ci en cinq groupes :
- les méthodes scalaires,
- les méthodes interactives,
- les méthodes floues,
- les méthodes exploitant une métaheuristique,
- les méthodes d'aide à la décision.

Les méthodes de ces cinq groupes peuvent aussi être rangées en trois familles de méthodes d'optimisation multiobjectif [Van Veldhuizen 99] :
- Les méthodes à préférence ***a priori*** :
  Dans ces méthodes, l'utilisateur définit le compromis qu'il désire réaliser (il fait part de ses préférences) avant de lancer la méthode d'optimisation.
  On retrouve dans cette famille la plupart des méthodes par agrégation (où les fonctions objectif sont fusionnées en une seule).
- Les méthodes à préférence ***progressive*** :
  Dans ces méthodes, l'utilisateur affine son choix de compromis au fur et à mesure du déroulement de l'optimisation.
  On retrouve dans cette famille les méthodes interactives.
- Les méthodes à préférence ***a posteriori*** :
  Dans ces méthodes, l'utilisateur choisit une solution de compromis en examinant toutes les solutions extraites par la méthode d'optimisation.
  Les méthodes de cette famille fournissent, à la fin de l'optimisation, une surface de compromis.

Il existe des méthodes d'optimisation multiobjectif qui n'entrent pas exclusivement dans une famille. Par exemple, on peut utiliser une méthode à préférence *a priori* en lui fournissant des préférences choisies au hasard. Le résultat sera alors un grand nombre de solutions qui seront présentées à l'utilisateur pour qu'il décide de la solution de compromis. Cette combinaison forme alors une méthode à préférence *a posteriori*.

Cette classification permet donc seulement de se faire une idée sur la démarche que l'on devra suivre pour obtenir notre résultat. Une classification plus "fine" est développée dans le chapitre 12.

1.13 Bibliographie commentée

[Ehrgott 97] Cet article présente différentes relations de dominance. Il est très mathématique mais reste un bon point de départ pour un tour d'horizon de ces relations.

[Hillier et al. 95] Ceci est un ouvrage anglophone d'introduction à la recherche opérationnelle. Tous les domaines de la recherche opérationnelle sont introduits à travers un exposé très didactique (programmation linéaire, théorie des jeux, programmation non linéaire). Beaucoup d'exemples bien commentés agrémentent l'exposé de chaque méthode.

[Miettinen 99] Ce livre, comme son titre l'indique, est entièrement dédié à l'optimisation multiobjectif. Les concepts de base y sont présentés à travers un exposé plutôt mathématique. Cet ouvrage se centre plus sur les méthodes *a priori* et *progressives*. Les principales caractéristiques mathématiques des méthodes sont présentées.

Chapitre 2

Les méthodes scalaires

2.1 La méthode de pondération des fonctions objectif

2.1.1 Principe

Cette approche de la résolution d'un problème d'optimisation multiobjectif est la plus évidente. D'ailleurs, on appelle aussi cette méthode l' "approche naïve" de l'optimisation multiobjectif [Coello 98]. Le but, ici, est de revenir à un problème d'optimisation monobjectif, dont il existe de nombreuses méthodes de résolution. La manière la plus simple de procéder consiste à prendre chacune des fonctions objectif, à leur appliquer un coefficient de pondération et à faire la somme des fonctions objectif pondérées. On obtient alors une nouvelle fonction objectif.

2.1.2 Présentation de la méthode

On part du problème P (voir page 18).
Le problème P se transforme de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & f_{eq}\left(\overrightarrow{x}\right) = \bigoplus_{i=1}^{k} w_i \cdot f_i\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

On a $\overrightarrow{x} \in \mathbb{R}^n$, $\overrightarrow{g}\left(\overrightarrow{x}\right) \in \mathbb{R}^m$ et $\overrightarrow{h}\left(\overrightarrow{x}\right) \in \mathbb{R}^p$.
Fréquemment, les coefficients de pondération respectent la relation suivante :
$w_i \geq 0$ pour tous les $i \in \{1, \cdots, k\}$ et :

$$\bigoplus_{i=1}^{k} w_i = 1 \qquad (2.1)$$

On peut représenter graphiquement le fonctionnement de la méthode sur un problème à deux objectifs. Le problème est le suivant :

$$\left|\begin{array}{ll}\text{minimiser} & f_1\left(\overrightarrow{x}\right)\\ \text{minimiser} & f_2\left(\overrightarrow{x}\right)\\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right)\leq 0\\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right)=0\end{array}\right.$$

Notre nouvelle fonction objectif aura pour expression :

$$f_{eq}\left(\overrightarrow{x}\right)=w_1\cdot f_1\left(\overrightarrow{x}\right)+w_2\cdot f_2\left(\overrightarrow{x}\right) \tag{2.2}$$

Ceci est l'expression d'une droite dans le plan $f_1,\ f_2$.

En effet, si l'on cherche à minimiser $f_{eq}\left(\overrightarrow{x}\right)$, on cherche en fait une constante C de l'équation de droite suivante la plus petite possible :

$$f_2\left(\overrightarrow{x}\right)=-\frac{w_1}{w_2}\cdot f_1\left(\overrightarrow{x}\right)+C \tag{2.3}$$

Cette équation de droite correspond à la courbe des isovaleurs de la fonction objectif équivalente.

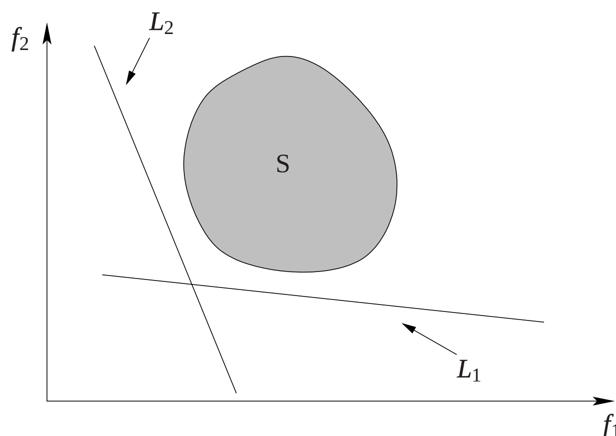


FIG. 2.1 – La méthode de pondération des fonctions objectif.

Sur la figure 2.1, l'ensemble S correspond à l'ensemble des valeurs du couple $(f_1,\ f_2)$ respectant les contraintes définies par $\overrightarrow{g}\left(\overrightarrow{x}\right)$ et $\overrightarrow{h}\left(\overrightarrow{x}\right)$. Les droites L_1 et L_2 correspondent à deux couples de coefficients de pondération $(w_1,\ w_2)$ différents.

Cette méthode consiste à "faire tangenter" la droite L_1 et la droite L_2 avec l'ensemble S. Le point de tangence est alors la solution recherchée.

Si l'on répète ce processus pour plusieurs valeurs des coefficients de pondération, les différentes solutions trouvées forment la surface de compromis.

Cette méthode n'est applicable qu'à des ensembles S convexes. Dans le cas contraire, elle ne permet pas de trouver la totalité de la surface de compromis. Par exemple, la portion en trait gras de la surface de compromis de la figure 2.2 ne peut être obtenue par cette méthode.

2.1.3 Deux exemples concrets

Exemple 1 : Le problème de "Schaffer's F2"

Considérons le problème suivant à deux fonctions objectif :

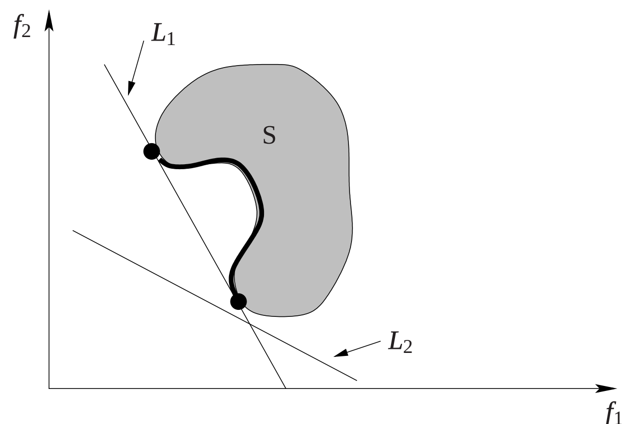


FIG. 2.2 – Une difficulté insurmontable pour la méthode de pondération des fonctions objectif.

$$\left| \begin{array}{ll} \text{minimiser} & f_1(x) = x^2 \\ \text{minimiser} & f_2(x) = (x-2)^2 \\ \text{avec} & x \in [0,2] \end{array} \right.$$

C'est un problème test appelé “Schaffer's F2”.

La représentation de l'ensemble solution de ce problème est différente de celle qui nous a servi à illustrer les différentes définitions de l'optimisation multiobjectif. En effet, si l'on cherche à exprimer f_2 en fonction de f_1, on obtient une équation d'ellipse :

$$\left| \begin{array}{l} f_1(\overrightarrow{x}) = x^2 \\ f_2(\overrightarrow{x}) = (x-2)^2 \quad = x^2 - 4 \cdot x + 4 \quad = f_1(\overrightarrow{x}) + 4 - 4 \cdot x \end{array} \right.$$

On obtient finalement :

$(f_2(\overrightarrow{x}) - f_1(\overrightarrow{x}) - 4)^2 = (4 \cdot x)^2$

$f_1^2 + f_2^2 - 2 \cdot f_1 \cdot f_2 - 8 \cdot f_1 - 8 \cdot f_2 + 16 = 0$

De plus, comme on le verra lors de la résolution du problème d'optimisation, la surface de compromis (qui correspond normalement à une surface) est ici une portion de courbe et l'ensemble des couples (f_1, f_2) est une courbe (une ellipse dans cet exemple).

Nous avons choisi cet exemple atypique car il permet d'illustrer numériquement et simplement le fonctionnement de certaines méthodes d'optimisation multiobjectif. Nous traitons dans l'exemple 2 un cas moins atypique de problème d'optimisation multiobjectif. Dans le reste du document, nous traiterons uniquement l'exemple 1 par souci de simplicité, en sachant que le même type de démarche pourra être appliqué à un problème d'optimisation multiobjectif classique.

La nouvelle fonction objectif s'écrit :

$$f_{eq}(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) \tag{2.4}$$

Les coefficients de pondération w_1 et w_2 respectent les conditions suivantes :

w_1 et $w_2 \geq 0$,

$w_1 + w_2 = 1$

On cherche le minimum de la fonction objectif $f_{eq}(x)$. Il vérifie les conditions suivantes nécessaires à l'existence d'un minimum :
$\frac{d\,f_{eq}(x)}{dx} = 0$,
$\frac{d^2\,f_{eq}(x)}{dx^2} > 0$
Cherchons les points de $f_{eq}(x)$ qui vérifient ces conditions :

$\frac{d\,f_{eq}(x)}{dx} = 2 \cdot x \cdot (w_1 + w_2) - 4 \cdot w_2$,

$\frac{d^2\,f_{eq}(x)}{dx^2} = 2 \cdot (w_1 + w_2) > 0$, car on a choisi $w_1 + w_2 = 1$.

Donc, le minimum se situe en :

$x^* = \frac{2 \cdot w_2}{w_1 + w_2} = 2 \cdot w_2$

On va maintenant calculer quatre points de la surface de compromis. On choisit quatre valeurs de w_1 variant de 0.2 à 0.8 par pas de 0.2. On en déduit les valeurs de w_2, puis on calcule x^*, $f_1(x^*)$ et $f_2(x^*)$. On constate que les valeurs de x^* respectent la contrainte, puis on réunit ces valeurs dans le tableau 2.1.

w_1	0.2	0.4	0.6	0.8
w_2	0.8	0.6	0.4	0.2
x^*	1.6	1.2	0.8	0.4
$f_1(x^*)$	2.56	1.44	0.64	0.16
$f_2(x^*)$	0.16	0.64	1.44	2.56

TAB. 2.1 – Tableau récapitulatif.

On trace maintenant ces points dans le plan f_1, f_2 et on les relie (voir figure 2.3).

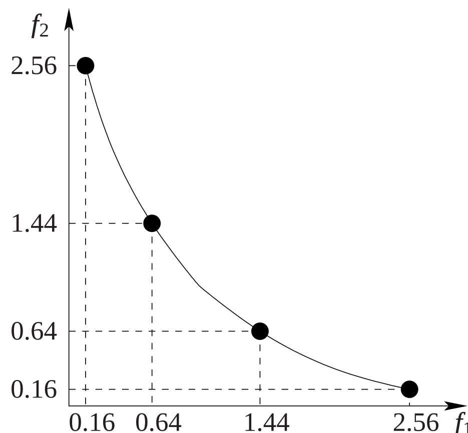


FIG. 2.3 – La surface de compromis obtenue pour le problème F2 de Schaffer.

On remarque que, dans cet exemple, on trouve une surface de compromis typique dans les problèmes de minimisation sur tous les objectifs. A la figure 2.4, on a représenté l'ensemble des valeurs réalisables du problème test, avec et sans la contrainte $x \in [0,2]$.

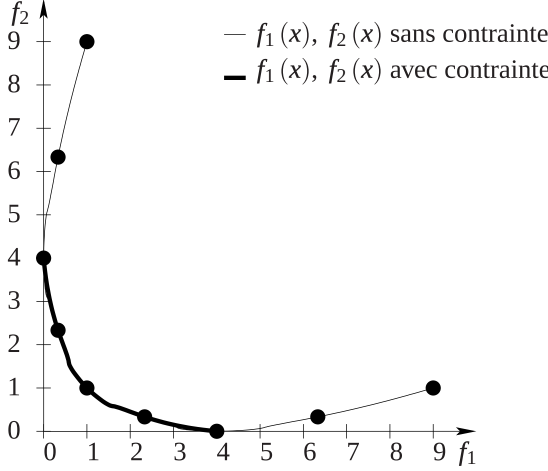


FIG. 2.4 – La surface de compromis totale, avec et sans la contrainte $x \in [0,2]$.

Exemple 2 : Le problème de "Binh"

Considérons le problème suivant à deux fonctions objectif :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(x_1,x_2) = x_1^2 + x_2^2 \\ \text{minimiser} & f_2(x_1,x_2) = (x_1-5)^2 + (x_2-5)^2 \\ \text{avec} & -5 \leq x_1 \leq 10,\ -5 \leq x_2 \leq 10 \end{array}\right.$$

Ce problème est appelé problème de "Binh". Un échantillon des solutions de ce problème est représenté à la figure 2.5.

On commence par former la nouvelle fonction objectif :

$$f_{eq}(x_1,x_2) = w_1 \cdot \left(x_1^2 + x_2^2\right) + w_2 \cdot \left((x_1+5)^2 + (x_2+5)^2\right)$$

avec
$w_1 + w_2 = 1$ et $w_1, w_2 \geq 0$.
On a alors :

$$f_{eq}(x_1,x_2) = x_1^2 + x_2^2 + 10 \cdot w_2 \cdot (x_1 + x_2) + 50 \cdot w_2$$

On cherche maintenant l'optimum de cette nouvelle fonction objectif :

$$\frac{\partial f_{eq}(x_1,x_2)}{\partial x_1} = 2 \cdot x_1 + 10 \cdot w_2 \tag{2.5}$$

$$\frac{\partial f_{eq}(x_1,x_2)}{\partial x_2} = 2 \cdot x_2 + 10 \cdot w_2 \tag{2.6}$$

$$\frac{\partial^2 f_{eq}(x_1,x_2)}{\partial x_1 \partial x_2} = \frac{\partial^2 f_{eq}(x_1,x_2)}{\partial x_1 \partial x_2} = 2 > 0 \tag{2.7}$$

L'équation 2.7 nous indique que la recherche des points qui annulent les équations 2.5 et 2.6 donnera un minimum de la fonction objectif. Donc :
$2 \cdot x_1^* + 10 \cdot w_2 = 0 \Rightarrow x_1^* = -5 \cdot w_2$
$2 \cdot x_2^* + 10 \cdot w_2 = 0 \Rightarrow x_2^* = -5 \cdot w_2$

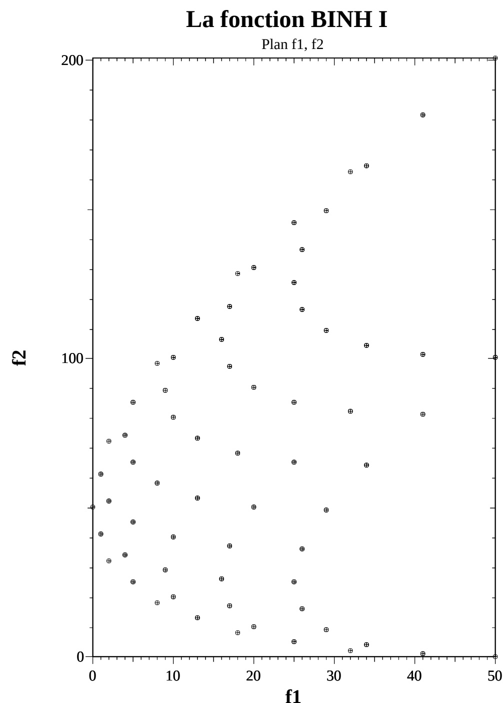


FIG. 2.5 – L'ensemble solution du problème de Binh.

On obtient alors les fonctions objectif suivantes :
$f_{eq}(w_2) = 50 \cdot w_2 \cdot (1 - w_2)$
$f_1(w_2) = 50 \cdot w_2^2$
$f_2(w_2) = 50 \cdot (1 - w_2)^2$
On peut alors calculer quelques solutions de la surface de compromis pour un certain nombre de valeurs du coefficient w_2. Ces résultats sont réunis dans le tableau 2.2.

w_2	$f_{eq}(w_2)$	$f_1(w_2)$	$f_2(w_2)$
0	0	0	50
0.1	4.5	0.5	40.5
0.2	8	2	32
0.3	10.5	4.5	24.5
0.4	12	8	18
0.5	12.5	12.5	12.5
0.6	12	18	8
0.7	10.5	24.5	4.5
0.8	8	32	2
0.9	4.5	40.5	0.5
1	0	50	0

TAB. 2.2 – Tableau récapitulatif.

L'allure de la surface de compromis est alors celle de la figure 2.6.

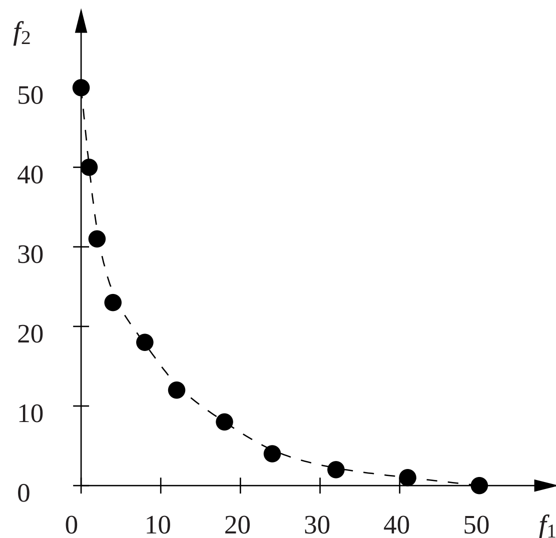


FIG. 2.6 – La surface de compromis du problème de Binh.

2.1.4 Discussion

Cette méthode a été l'une des premières utilisées. D'un point de vue algorithmique, elle est très efficace. On peut, en faisant varier les coefficients de pondération, retrouver la surface de compromis, si le domaine réalisable est convexe.

Cependant, cette méthode ne permet pas de trouver les solutions enfermées dans une concavité. Dans certains problèmes d'optimisation de structures mécaniques, elle peut se comporter de manière catastrophique [Koski 85].

2.2 La méthode de Keeney-Raiffa

2.2.1 Principe

Cette méthode utilise le produit des fonctions objectif pour se ramener à un problème d'optimisation monobjectif. L'approche utilisée ici est semblable à celle utilisée dans la méthode de pondération des fonctions objectif. La fonction objectif ainsi obtenue s'appelle la fonction d'utilité de Keeney-Raiffa.

On retrouve cette fonction dans la théorie de la fonction utilité multi-attribut exposée dans [Keeney et al. 93] (M.A.U.T, *Multi Attribute Utility Theory*). Cette théorie, issue des sciences d'aide à la décision, traite des propriétés et de la manière de créer une “fonction d'utilité”.

2.2.2 Présentation de la méthode

On part du problème P (voir page 18).

On transforme la partie “objectifs” en utilisant la fonction suivante :

$$f_{eq}\left(\overrightarrow{x}\right) = K \cdot \prod_{i=1}^{k} \left(k_i \cdot u_i\left(f_i\left(\overrightarrow{x}\right)\right) + 1\right) \tag{2.8}$$

Dans cette expression,

k_i est un coefficient de normalisation (compris entre 0 et 1).

$u_i(\overrightarrow{y})$ est une fonction qui est strictement non décroissante en fonction de y et qui peut incorporer des non-linéarités. Par exemple, $u_i(y) = y^2$. C'est une fonction strictement non décroissante de y si $y \in \mathbb{R}^+$.

Si l'on prend $u_i(y) = y^2$, $K = 1$ et $k_i = 1$, le problème P se transforme de la manière suivante :

$$\left| \begin{array}{ll} \text{minimiser} & f_{eq}(\overrightarrow{x}) = \prod_{i=1}^{k} \left((f_i(\overrightarrow{x}))^2 + 1 \right) \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array} \right.$$

2.3 La méthode de la distance à un objectif de référence

2.3.1 Principe

Cette méthode permet de transformer un problème d'optimisation multiobjectif en un problème d'optimisation monobjectif.

La somme que l'on va utiliser ici prendra la forme d'une distance ($\sqrt[k]{\oplus()^k}$ ou la racine k^e de la somme d'éléments élevés à une certaine puissance k). De plus, dans certains cas on normalisera les éléments par rapport à une valeur avant d'en faire la somme [Azarm 96, Miettinen 99].

2.3.2 Présentation de la méthode

On part encore du problème P.

Dans les définitions suivantes, le vecteur $\overrightarrow{F}$ (de coordonnées F_i, $i \in \{1, \cdots, k\}$) est un vecteur dont les coordonnées correspondent à celles d'un objectif idéal (ou de référence) au sens défini par l'utilisateur (cela peut être le point idéal, ou un autre point qui représente mieux les préférences de l'utilisateur). Le vecteur $\overrightarrow{F}$ est choisi par l'utilisateur.

On peut, par exemple, utiliser la distance absolue. Voici la définition de cette distance :

$$L_r\left(\overrightarrow{f}(\overrightarrow{x})\right) = \left[\bigoplus_{i=1}^{k} |F_i - f_i(\overrightarrow{x})|^r \right]^{\frac{1}{r}} \quad (2.9)$$

avec $1 \leq r < \infty$.

Par exemple :

– $r = 1$:

$$L_1\left(\overrightarrow{f}(\overrightarrow{x})\right) = \bigoplus_{i=1}^{k} |F_i - f_i(\overrightarrow{x})| \quad (2.10)$$

– $r = 2$:

$$L_2\left(\overrightarrow{f}(\overrightarrow{x})\right) = \left[\bigoplus_{i=1}^{k} |F_i - f_i(\overrightarrow{x})|^2 \right]^{\frac{1}{2}} \quad (2.11)$$

– $r \to \infty$:

$$L_\infty\left(\overrightarrow{f}\left(\overrightarrow{x}\right)\right) = \max_{i\in\{1\cdots,k\}} \left(F_i - f_i\left(\overrightarrow{x}\right)\right) \tag{2.12}$$

Pour cette dernière forme, la méthode se transforme en la méthode dite “min-max” (ou méthode de Tchebychev).

On peut aussi utiliser la distance relative. Voici la définition de cette distance :

$$L_r^{Rel}\left(\overrightarrow{f}\left(\overrightarrow{x}\right)\right) = \left[\bigoplus_{i=1}^{k}\left|\frac{F_i - f_i\left(\overrightarrow{x}\right)}{F_i}\right|^r\right]^{\frac{1}{r}} \tag{2.13}$$

avec $1 \leq r < \infty$.

Par exemple, la méthode dite “min-max” relatif est obtenue pour $r \to \infty$:

$$L_{minmax}\left(\overrightarrow{f}\left(\overrightarrow{x}\right)\right) = \max_{i\in\{1,\cdots,k\}} \left(\frac{F_i - f_i\left(\overrightarrow{x}\right)}{F_i}\right) \tag{2.14}$$

Pour finir, on peut aussi utiliser des coefficients de pondération dans l’expression de la distance. Ces coefficients de pondération permettent de chercher un point dans une certaine direction, comme avec la méthode de pondération des fonctions objectif. On obtient alors l’expression suivante :

$$L_r^{W}\left(\overrightarrow{f}\left(\overrightarrow{x}\right)\right) = \left[\bigoplus_{i=1}^{k}\left|w_i\cdot\left(F_i - f_i\left(\overrightarrow{x}\right)\right)\right|^r\right]^{\frac{1}{r}}$$

avec $1 \leq r \leq \infty$.

Pour la méthode “min-max”, on obtient :

$$L_{minmax}\left(\overrightarrow{f}\left(\overrightarrow{x}\right)\right) = \max_{i\in\{1,\cdots,k\}} \left(w_i\cdot\left(F_i - f_i\left(\overrightarrow{x}\right)\right)\right)$$

Appliquons la méthode de la distance au problème P.

Pour transformer notre problème, on va utiliser la distance L_2. On aura alors le problème suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_{eq}\left(\overrightarrow{x}\right) = \sqrt{\oplus_{i=1}^{k}\left|F_i - f_i\left(\overrightarrow{x}\right)\right|^2} \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

On peut souligner que, lorsque l’on utilise ces méthodes pour la recherche de solutions optimales au sens de Pareto, il n’est pas nécessaire d’ajouter le terme $\sqrt{}$ dans l’expression de $f_{eq}\left(\overrightarrow{x}\right)$ car cet opérateur est monotone. Ceci autorise un certain gain de temps dans le calcul de la fonction objectif équivalente.

La figure 2.7 illustre le fonctionnement de cette méthode.

Sur cette figure, on a représenté notre point de référence (qui est différent du point idéal) par un point noir. La surface de compromis, elle, est représentée par un trait noir épais. Comme on peut le voir à la figure 2.10, le choix de ce point de référence influe beaucoup sur la “qualité” du résultat. Cette méthode permet donc de trouver la surface de compromis vue d’un certain point de référence.

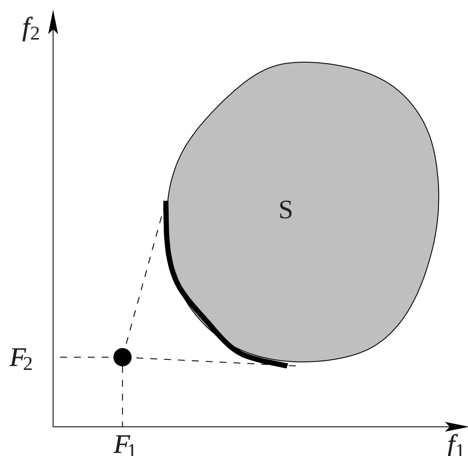


FIG. 2.7 – La méthode de la distance.

2.3.3 Les courbes isovaleur

Pour tracer les courbes isovaleur de ces méthodes d'agrégation des fonctions objectif, il faut procéder de la même manière qu'avec la méthode de pondération des fonctions objectif : minimiser $f_{eq}\left(\overrightarrow{x}\right)$ revient à trouver une valeur de la variable $\overrightarrow{x}$ telle que la constante C de l'expression suivante soit minimale :

$$C = w_1 \cdot \left(f_1\left(\overrightarrow{x}\right) - F_1\right)^2 + w_2\left(f_2\left(\overrightarrow{x}\right) - F_2\right)^2 \tag{2.15}$$

C'est l'expression d'une ellipse de centre (F_1, F_2), de "petit" axe $2 \cdot \sqrt{\frac{C}{w_1}}$ et de "grand" axe $2 \cdot \sqrt{\frac{C}{w_2}}$.

Le but va donc être de trouver le plus petit C tel que l'ellipse vienne "tangenter" l'ensemble des solutions S. Cette courbe isovaleur est représentée à la figure 2.8.

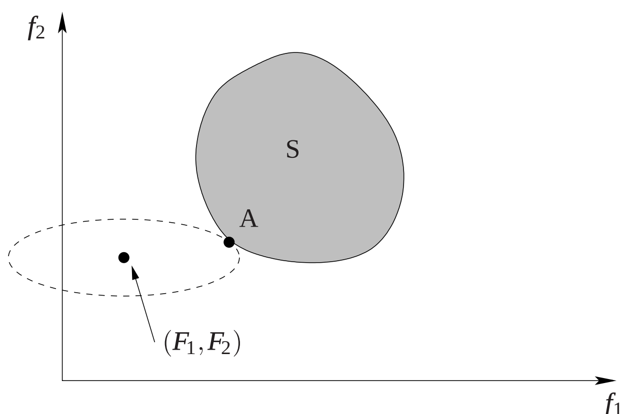


FIG. 2.8 – Une courbe isovaleur de la méthode de la distance.

Pour la méthode de Tchebychev, l'expression de la courbe isovaleur est encore plus simple :

$$\begin{cases} f_1\left(\overrightarrow{x}\right) = C & \text{si} \quad w_1\left(f_1\left(\overrightarrow{x}\right) - F_1\right) > w_2\left(f_2\left(\overrightarrow{x}\right) - F_2\right) \\ f_1\left(\overrightarrow{x}\right) = C & \text{si} \quad w_1\left(f_1\left(\overrightarrow{x}\right) - F_1\right) < w_2\left(f_2\left(\overrightarrow{x}\right) - F_2\right) \end{cases} \tag{2.16}$$

Cette courbe est représentée à la figure 2.9.

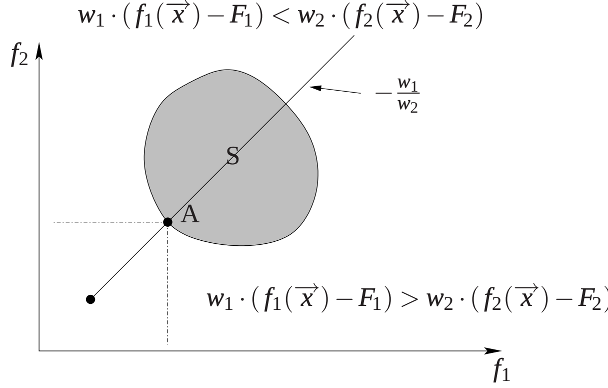


FIG. 2.9 – Une courbe isovaleur de la méthode de Tchebychev.

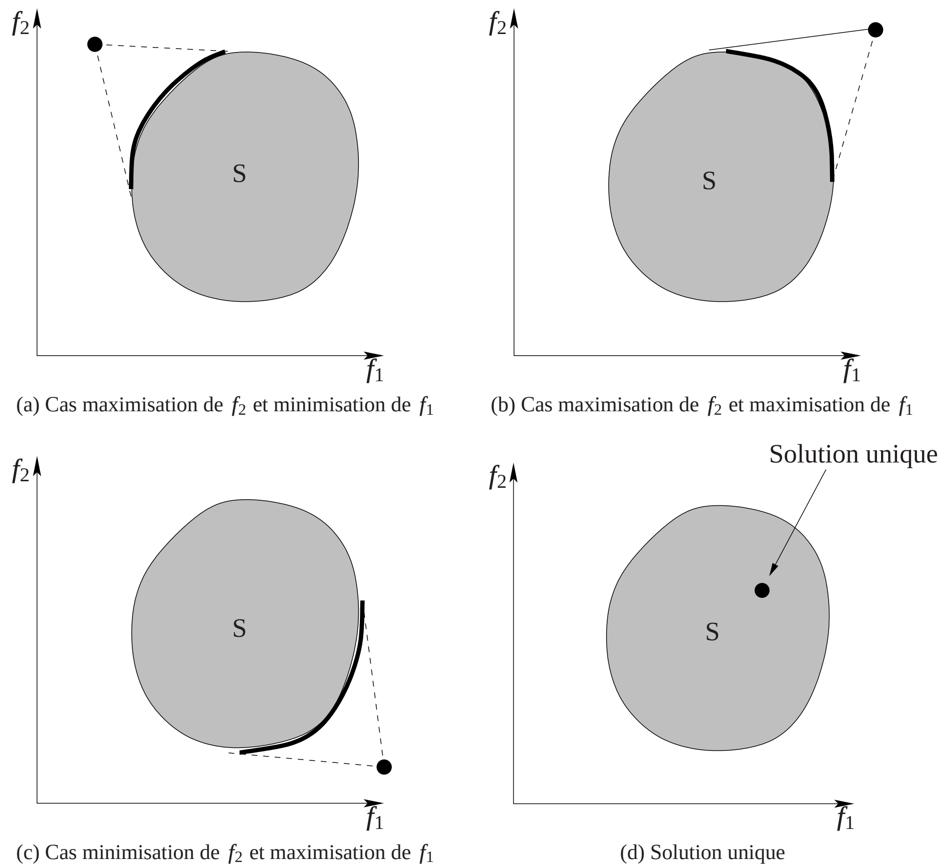
2.3.4 Discussion

Pour que cette méthode fonctionne bien, on doit être capable de construire un bon point de référence. Sinon, les solutions que l'on trouvera ne seront pas forcément optimales, dans le sens où elles ne répondront pas nécessairement au problème initial.

La figure 2.10 illustre le défaut de fonctionnement de cette méthode, lorsque l'objectif idéal est mal choisi. On cherche à minimiser les deux objectifs (f_1 et f_2) ; le bon positionnement du point de référence est indiqué à la figure 2.7.

- A la figure 2.10-a, le fait d'avoir positionné le point de référence dans le coin supérieur gauche nous donne une surface de compromis équivalente à un problème pour lequel on minimise l'objectif f_1 et on maximise l'objectif f_2.
- A la figure 2.10-b, le fait d'avoir positionné le point de référence dans le coin supérieur droit nous donne une surface de compromis équivalente à un problème pour lequel on maximise les objectifs f_1 et f_2.
- A la figure 2.10-c, le fait d'avoir positionné le point de référence dans le coin inférieur droit nous donne une surface de compromis équivalente à un problème pour lequel on maximise l'objectif f_1 et on minimise l'objectif f_2.
- Pour finir, à la figure 2.10-d, le fait d'avoir positionné le point de référence à l'intérieur de l'ensemble des valeurs du couple (f_1, f_2) nous donne une solution unique qui ne correspond ni à une maximisation, ni à une minimisation sur chacune des fonctions objectif.

De plus, comme la méthode de pondération des fonctions objectif, cette méthode ne permet de trouver des points cachés dans des non-convexités que sous certaines conditions, présentées dans [Messac et al. 00].



(a) Cas maximisation de f_2 et minimisation de f_1

(b) Cas maximisation de f_2 et maximisation de f_1

(c) Cas minimisation de f_2 et maximisation de f_1

(d) Solution unique

FIG. 2.10 – La difficulté du choix d'un objectif idéal.

2.4 La méthode du compromis

2.4.1 Principe

Après avoir étudié les méthodes qui permettent de fusionner les fonctions objectif en une seule (on parle aussi d' "agrégation" des fonctions objectif), nous allons étudier des méthodes qui permettent de transformer un problème d'optimisation multiobjectif en un problème d'optimisation monobjectif comportant quelques contraintes supplémentaires.

La démarche est la suivante :

- on choisit un objectif à optimiser prioritairement ;
- on choisit un vecteur de contraintes initial ;
- on transforme le problème en conservant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalité.

On appelle aussi cette méthode la méthode de la contrainte [Miettinen 99].

2.4.2 Présentation de la méthode

On part du problème P.

On suppose que la fonction objectif prioritaire est la fonction de rang 1.

On choisit un vecteur de contraintes $_i$, $i \in \{2, \cdots, k\}$, $_i \geq 0$.
On transforme le problème P de la manière suivante :

$$\left| \begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) \\ \text{avec} & f_2\left(\overrightarrow{x}\right) \leq \ _2 \\ & \vdots \\ & f_k\left(\overrightarrow{x}\right) \leq \ _k \\ & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array} \right.$$

A la figure 2.11, on a représenté le fonctionnement de cette méthode pour un problème à deux objectifs.

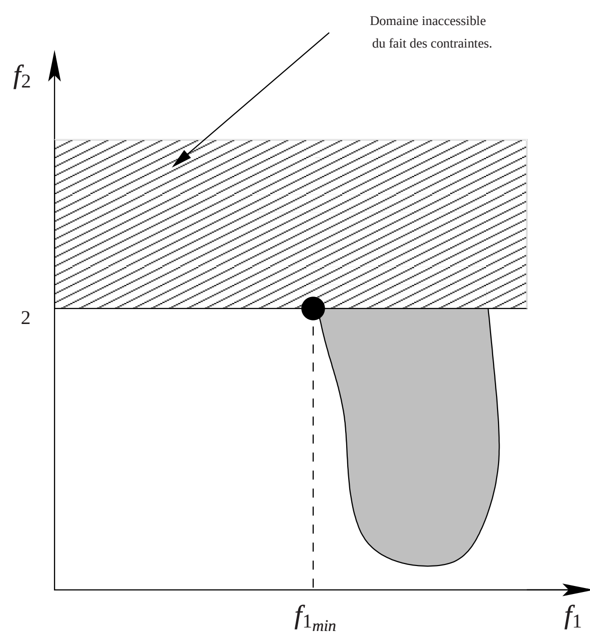


FIG. 2.11 – La méthode du compromis.

2.4.3 Un exemple concret

On reprend l'exemple du problème "Schaffer's F2".
En appliquant la méthode décrite précédemment, on obtient le problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & f_1(x) = x^2 \\ \text{avec} & f_2(x) = (x-2)^2 \leq \\ & x \in [0,2] \end{array} \right.$$

On décide maintenant de résoudre ce problème en choisissant quatre valeurs pour la variable : 0, 1, 2, 3.

On commence par réécrire la contrainte :

$$(x-2)^2 \leq \quad \Leftrightarrow 2-\sqrt{\ } \leq x \leq 2+\sqrt{\ }$$

Avec les valeurs de que nous avons choisies :

$$0 < 2-\sqrt{\ } \leq 2$$

Lorsque l'on met en relation cette expression avec $x \in [0,2]$, on obtient la nouvelle contrainte :

$$x \in [2-\sqrt{\ },2]$$

	Contrainte	Solution x^*	$f_1(x^*)$	$f_2(x^*)$
0	$x \in [2,2]$	2	4	0
1	$x \in [1,2]$	1	1	1
2	$x \in [2-\sqrt{2},2]$	$2-\sqrt{2}$	$6-4\cdot\sqrt{2}$	2
3	$x \in [2-\sqrt{3},2]$	$2-\sqrt{3}$	$7-4\cdot\sqrt{3}$	3

On représente les différentes solutions dans le plan (f_1, f_2) (voir figure 2.12).

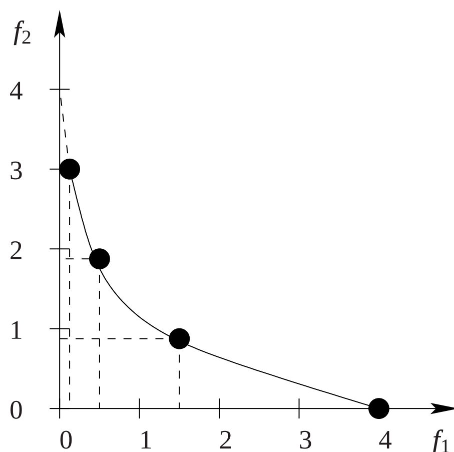


FIG. 2.12 – Représentation de la surface de compromis pour le problème Schaffer's F2, en choisissant les valeurs 0,1,2 et 3 pour .

A la figure 2.13, on a représenté la démarche qui a été suivie pour résoudre le problème test. On voit, sur cette figure, l'effet de la modification de la borne de notre nouvelle contrainte. De plus, on voit apparaître, pas à pas, les solutions de notre problème et, par la même occasion, la surface de compromis.

Pour des problèmes simples, le choix des valeurs de $_i$ peut donner une bonne répartition des solutions sur la surface de compromis. En revanche, dans la plupart des cas concrets, le choix (arbitraire) des valeurs de $_i$ ne permet pas d'obtenir une bonne répartition des solutions sur la surface de compromis (c'est le cas pour notre exemple dans lequel il manque un point entre les valeurs 2 et 3 de la coordonnée f_1). Alors, on peut ne pas avoir assez de points et rencontrer de grandes difficultés pour extrapoler l'allure de la surface de compromis.

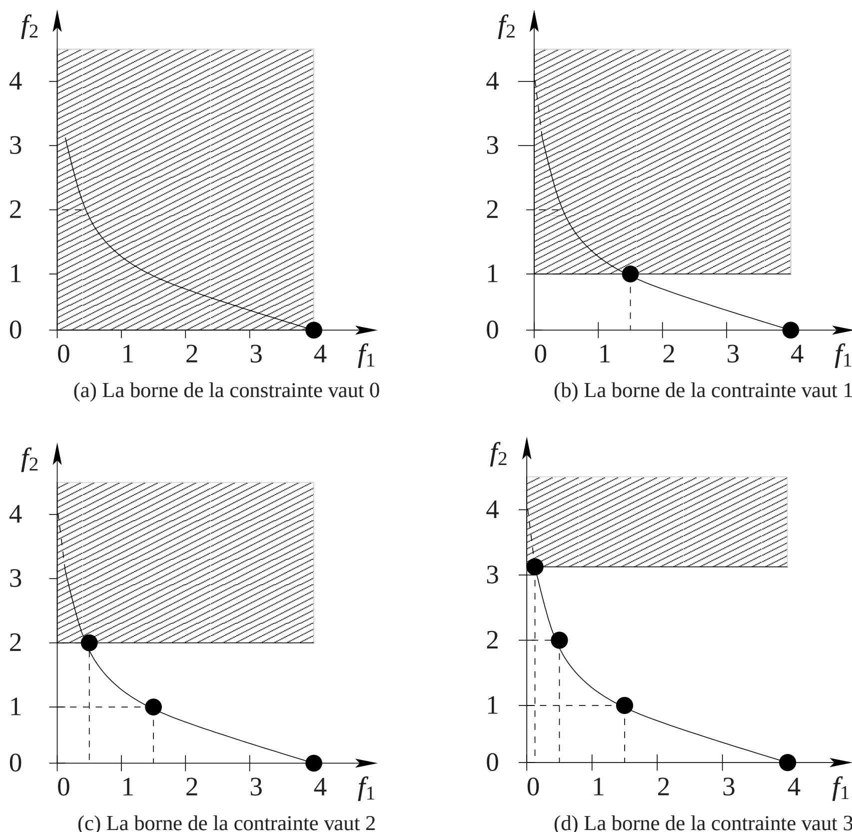


FIG. 2.13 – La résolution pas à pas du problème test.

2.4.4 Discussion

Dans [Eschenauer et al. 90], on montre que cette méthode peut se ramener à la méthode de pondération des fonctions objectif moyennant une légère transformation.

Les principaux inconvénients de cette méthode sont les suivants : elle est gourmande en temps de calcul et la programmation de l'algorithme peut être extrêmement difficile s'il y a trop de fonctions contraintes.

Cependant, la relative simplicité de l'énoncé de la méthode l'a rendue populaire.

2.5 Les méthodes hybrides

2.5.1 Principe

Comme nous allons le voir, il est possible d'exploiter plusieurs méthodes pour en former une nouvelle. Dans ce cas, nous obtenons une "méthode hybride".

2.5.2 Présentation d'une méthode hybride

La méthode hybride la plus connue est la méthode de Corley. Cette méthode utilise la méthode de pondération des fonctions objectif et la méthode du compromis.

On part du problème P. On le transforme de la manière suivante :

$$
\left|\begin{array}{ll}
\text{minimiser} & \oplus_{i=1}^{k} w_i \cdot f_i\left(\overrightarrow{x}\right) \\
\text{avec} & f_j\left(\overrightarrow{x}\right) \leq {}_j,\ j = 1, \cdots, k \\
 & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\
\text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0
\end{array}\right.
$$

A la figure 2.14, on a représenté le fonctionnement de cette méthode pour un problème à deux objectifs.

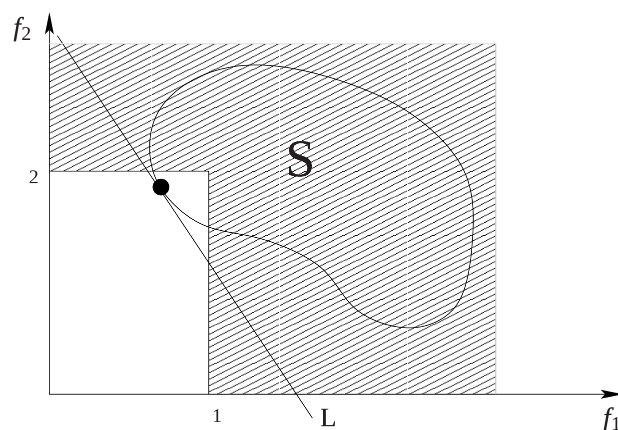


FIG. 2.14 – La méthode hybride de Corley.

Sur cette figure, on peut voir que l'ajout des deux contraintes ($f_1\left(\overrightarrow{x}\right) \leq {}_1$ et $f_2\left(\overrightarrow{x}\right) \leq {}_2$) a permis de restreindre l'espace de recherche. Ensuite, on applique la méthode de pondération des fonctions objectif sur cet espace restreint de recherche. Pour un vecteur des poids $\overrightarrow{w} = (w_1, w_2)$ auquel on fait correspondre la droite L, chercher une valeur optimale revient à faire tangenter la droite L avec l'espace restreint de recherche.

2.5.3 Discussion

Cette méthode permet de combiner les avantages des deux méthodes citées précédemment (la méthode de pondération des fonctions objectif et la méthode du compromis). Elle est ainsi efficace sur différents types de problèmes, qu'ils soient convexes ou non convexes. Cependant, une difficulté survient : le nombre de paramètres à déterminer a été multiplié par deux. Il sera beaucoup plus difficile à l'utilisateur d'exprimer sa préférence en jouant sur l'ensemble des paramètres. Pour plus d'informations, voir [Miettinen 99].

2.6 La méthode dite du “but à atteindre”

2.6.1 Principe

Dans la littérature anglo-saxonne, on désigne cette méthode sous l’appellation suivante : *goal attainment method*.

Contrairement à la méthode de pondération des fonctions objectif, cette méthode n’impose pas de travailler sur un domaine réalisable convexe.

La démarche suivie est la suivante :

- on choisit un vecteur de fonctions objectif initial $\overrightarrow{F}$;
- on choisit une direction de recherche (on fournit, en quelque sorte, des coefficients de pondération, comme pour la méthode de pondération des fonctions objectif) $\overrightarrow{w}$;
- on cherche ensuite à minimiser un coefficient scalaire qui représente l’écart par rapport à l’objectif initial $\overrightarrow{F}$ que l’on s’était fixé.

2.6.2 Présentation de la méthode

On part du problème P.

On choisit un vecteur de fonctions objectif initial $\overrightarrow{F} \in \mathbb{R}^k$,

un ensemble de coefficients w_i, $i \in \{1, \cdots, k\}$,

et l’on désigne par un nombre scalaire que l’on cherchera à minimiser.

Le problème P devient alors :

$$
\left|
\begin{array}{ll}
\text{minimiser} & \\
\text{avec} & f_1\left(\overrightarrow{x}\right) - w_1 \cdot \quad \leq F_1 \\
 & \vdots \\
 & f_k\left(\overrightarrow{x}\right) - w_k \cdot \quad \leq F_k \\
 & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\
\text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0
\end{array}
\right.
$$

La figure 2.15 représente le fonctionnement de cette méthode, en considérant un problème d’optimisation à deux objectifs.

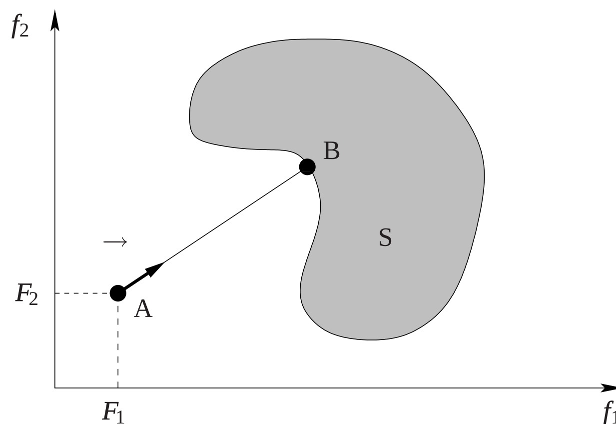


FIG. 2.15 – La méthode du but à atteindre, pour un problème à deux objectifs.

Sur cette figure, le point A représente l'objectif de référence, le vecteur $\overrightarrow{w}$ représente une direction de recherche et le point B représente le but à atteindre.

Le but de cette méthode sera donc de trouver une valeur (qui représente la distance entre le point A et B) telle que le point B réside sur la frontière de l'ensemble S et que la distance entre le point A et B soit minimisée. De plus, on voit bien sur ce graphique que cette méthode est efficace lorsque l'on doit traiter des ensembles non convexes.

Souvent, on appelle la variable le degré de sur ou sous-dépassement de l'objectif $\overrightarrow{F}$.

Pour obtenir la surface de compromis, il faudra prendre, comme précédemment, plusieurs directions de recherche (et peut-être même plusieurs vecteurs de fonctions objectif initiaux différents).

Pour plus d'informations, voir [Pentzaropoulos et al. 93].

2.6.3 Un exemple concret

Considérons à nouveau le problème "Schaffer's F2". On transforme le problème de la manière suivante :

$$\begin{array}{|ll} \text{minimiser} & \\ \text{avec} & x^2 - w_1 \cdot \quad \leq 1 \\ & (x-2)^2 - w_2 \cdot \quad \leq 1 \\ \text{et} & x \in [0,2] \end{array}$$

Notre connaissance du problème nous a fait choisir $F_1 = 1$ et $F_2 = 1$. De plus, on s'impose $w_1 + w_2 = 1$ et $w_1, w_2 \geq 0$.

Nous allons maintenant chercher un $_{min}$ tel que les contraintes seront respectées. Pour mieux visualiser notre but, nous représentons à la figure 2.16 la fonction $(g_1(f_1, f_2, \ ), g_2(f_1, f_2, \ ),$ $)$ dans le plan $(f_1, f_2, \ )$. Soient les notations suivantes :

– on appelle Δ_1 la droite $g_1(f_1, f_2, \ ) = f_1(x) - w_1 \cdot \ $,
– on appelle Δ_2 la droite $g_2(f_1, f_2, \ ) = f_2(x) - w_2 \cdot \ $.

Le point intéressant dans cette figure est A. En effet, pour cette valeur de $_A$, les contraintes de notre problème sont vérifiées quel que soit x. Ce point est défini par :

$$\begin{cases} g_1(f_1, f_2, \ _A) = 1 \\ g_2(f_1, f_2, \ _A) = 1 \end{cases}$$

On résout ce problème (on élimine $_A$ et on trouve x) :

$$\begin{cases} x^2 - w_1 \cdot \ _A = 1 \\ (x-2)^2 - w_2 \cdot \ _A = 1 \end{cases}$$

$$(x-2)^2 - \frac{w_2}{w_1} \cdot \left(x^2 - 1\right) = 1$$

$$x^2 \cdot \left(1 - \frac{w_2}{w_1}\right) - 4 \cdot x + 3 + \frac{w_2}{w_1} = 0$$

On a alors :

– $\Delta = 4 + 8 \cdot \frac{w_2}{w_1} + 4 \cdot \left(\frac{w_2}{w_1}\right)^2$ ou

$\Delta = 4 \cdot \left(1 + \frac{w_2}{w_1}\right)^2$ donc $\sqrt{\Delta} = 2 \cdot \left(1 + \frac{w_2}{w_1}\right)$

– $x_1 = \frac{4+\sqrt{\Delta}}{2 \cdot \left(1 - \frac{w_2}{w_1}\right)}$, $x_2 = \frac{4-\sqrt{\Delta}}{2 \cdot \left(1 - \frac{w_2}{w_1}\right)} = 1, \forall w_1, w_2$

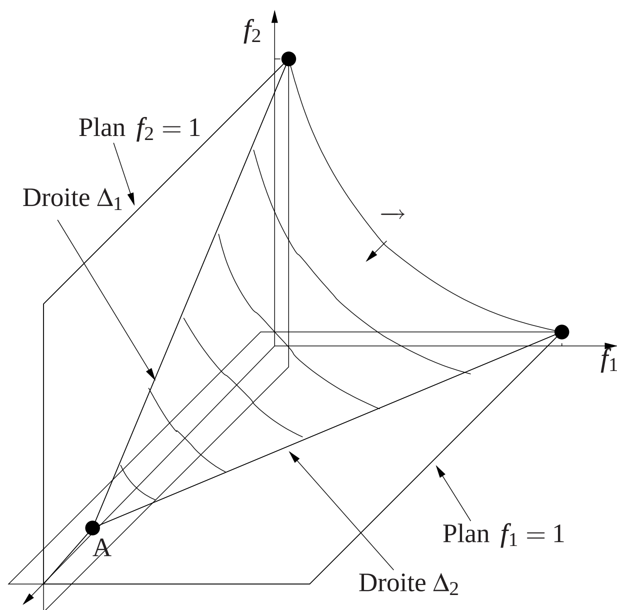


FIG. 2.16 – Représentation du problème dans le plan (f_1, f_2,).

w_1	w_2	Δ	x_1	x	f_1	f_2	A
0.05	0.95	1600	1.78	1.78	3.17	0.04	43.37
0.1	0.9	400	1.5	1.5	2.25	0.25	12.5
0.15	0.85	178	1.14	1.14	1.3	0.74	1.99
0.2	0.8	100	0.67	0.67	0.44	1.77	−2.75
0.21	0.79	90.7	0.55	0.55	0.30	2.10	−3.32
0.22	0.78	82.6	0.42	0.42	0.17	2.49	−3.74
0.23	0.77	75.6	0.29	0.29	0.08	2.92	−3.98
0.24	0.76	69.44	0.15	0.15	0.02	3.42	−4.07
0.25	0.75	64	0	0	0	4	−4
0.3	0.7	44.44	−1	1	1	1	0
0.4	0.6	25	−6	1	1	1	87.5
0.5	0.5	16	×	1	1	1	×
0.6	0.4	11.11	14	1	1	1	325
0.7	0.3	8.16	9	1	1	1	114
0.8	0.2	6.25	7.33	1	1	1	65.9
0.9	0.1	4.93	6.5	1	1	1	45.83

TAB. 2.3 – Les résultats du problème concret.

Il ne nous reste plus qu'à remplir le tableau 2.3.

Dans ce tableau, les différentes variables correspondent à :

w_1 et w_2 : les coefficients de pondération.

Δ : le discriminant du polynôme.

x : les valeurs de x_1 et/ou x_2 qui respectent la contrainte $x \in [0, 2]$. Pour les neuf premières valeurs, x_1 a été choisi et pour les dernières valeurs, x_2 (seul possible) a été choisi. De cette manière, on a pu assurer une certaine diversité dans la répartition des points sur la surface de compromis.

$f_1(x)$: la valeur de la fonction objectif f_1 pour la valeur de x correspondante.

$f_2(x)$: la valeur de la fonction objectif f_2 pour la valeur de x correspondante.

$_A(x)$: la valeur de $_A$ pour la valeur de x correspondante. Cette valeur est calculée en utilisant la formule suivante :

$$_A = \frac{x^2 - 1}{w_1}$$

On représente les dix premières solutions obtenues dans le plan (f_1, f_2) (voir la figure 2.17).

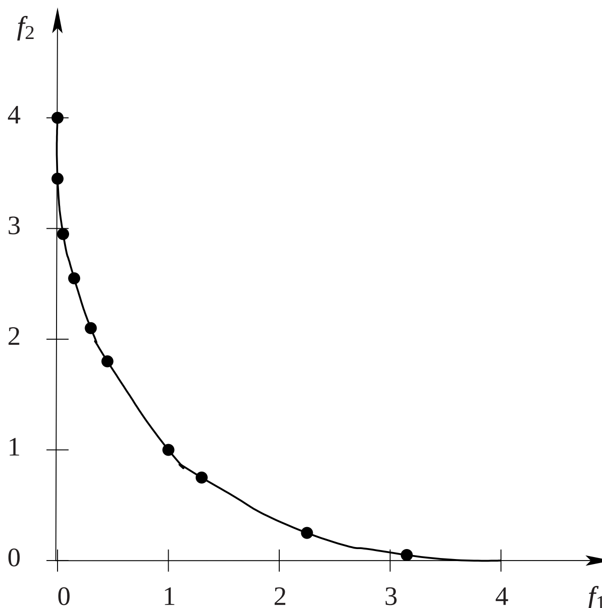


FIG. 2.17 – Représentation des solutions dans le plan (f_1, f_2) obtenues avec la méthode du but à atteindre.

Comme on peut le voir sur cet exemple, la représentation des solutions permet de tracer la surface de compromis du problème d'optimisation. En revanche, on peut remarquer que les solutions sont obtenues avec des valeurs de poids w_1, w_2 comprises dans une plage de variation assez étroite. Par rapport à la méthode de pondération des fonctions objectif, on peut dire que la méthode du but à atteindre est beaucoup plus sensible dans le choix des poids servant à obtenir une solution. Le problème est que, dans cette méthode, les poids ne reflètent pas les préférences de l'utilisateur. Ces poids n'ont aucune signification "physique", comme avec la méthode de pondération des fonctions objectif.

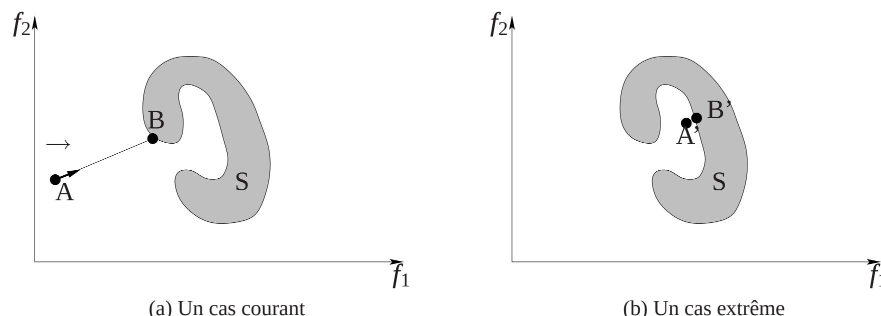
Le choix des coefficients de pondération uniformément répartis dans l'intervalle $[0, 1]$ ne nous permettrait pas d'obtenir une représentation uniforme des solutions sur la surface de compromis. Pour obtenir une répartition uniforme, il faut procéder de la manière suivante : faire un premier "passage" pour repérer la zone d'intérêt, puis se focaliser sur cette zone en faisant un second "passage", qui permettra d'affiner les poids, pour obtenir une meilleure répartition des points sur la surface de compromis.

2.6.4 Discussion

Le principal avantage de cette méthode est qu’elle permet de travailler avec un ensemble de solutions *S* pas nécessairement convexe. Cependant, il peut exister des formes de domaines pour lesquelles la solution dépend du choix du but initial. La figure 2.18-b représente un exemple d’un tel domaine. Heureusement, ce cas est extrême.

Cette conclusion peut être tirée pour tous les types de méthodes basées sur une distance.

De plus, lors de la résolution de l’exemple, si nous n’avions pas fait une deuxième optimisation en réglant mieux nos poids, nous aurions rencontré une difficulté typique de l’optimisation multiobjectif : la non-uniformité de la représentation des solutions sur la surface de compromis.



(a) Un cas courant

(b) Un cas extrême

FIG. 2.18 – Une difficulté pour la méthode du but à atteindre.

A et A’ :	objectifs initiaux.
B et B’ :	solutions optimales.
$\overrightarrow{w}$ et $\overrightarrow{w}'$:	directions de recherche.

2.7 La méthode dite du “but programmé”

2.7.1 Principe

Cette méthode (en anglais *Goal programming*) se rapproche beaucoup de la méthode du but à atteindre (en anglais *Goal attainment*).

La principale différence est qu’après avoir transformé la forme du problème d’optimisation, on se retrouve avec des contraintes d’égalité et non plus des contraintes d’inégalité.

La démarche est la suivante :

- on choisit un vecteur de fonctions objectif initial $\overrightarrow{F}$;
- à chaque objectif, on associe deux nouvelles variables que l’on appelle les “déviations” par rapport au vecteur de fonctions objectif initial fixé ;
- on minimise ensuite une des deux variables. Le choix de la variable se fait en fonction du type de dépassement que l’on veut (au-dessus ou au-dessous de l’objectif que l’on s’est fixé).

2.7.2 Présentation de la méthode

On part du problème P.

On choisit un vecteur de fonctions objectif initial $\overrightarrow{F} \in \mathbb{R}^k$.

On associe aussi un ensemble de variables d_i^+ et d_i^- à chaque fonction objectif $f_i(\overrightarrow{x})$, $i \in \{1, \cdots, k\}$ (les "déviations").

On obtient alors le problème suivant :

$$
\left|\begin{array}{ll}
\text{minimiser} & \left(d_1^+ \text{ ou } d_1^-, \cdots, d_k^+ \text{ ou } d_k^-\right) \\
\text{avec} & f_1(\overrightarrow{x}) = F_1 + d_1^+ - d_1^- \\
 & \vdots \\
 & f_k(\overrightarrow{x}) = F_k + d_k^+ - d_k^- \\
 & \overrightarrow{h}(\overrightarrow{x}) = 0 \\
\text{et} & \overrightarrow{g}(\overrightarrow{x}) \leq 0
\end{array}\right.
$$

Les variables de déviation que l'on cherche à minimiser doivent respecter certaines contraintes :
d_i^+ et $d_i^- \geq 0$,
$d_i^+ \cdot d_i^- = 0$ avec $i \in \{1, \cdots, k\}$.

En fonction de la façon dont on désire atteindre le but $\overrightarrow{F}$, on cherchera à minimiser différentes combinaisons des coefficients d_i^- et d_i^+. Ces combinaisons sont réunies dans le tableau 2.4.

Type	Valeur de la déviation	Variable
On désire atteindre par valeurs supérieures l'objectif i	Positive	d_i^+
On désire atteindre par valeurs inférieures l'objectif i	Négative	d_i^-
On désire atteindre l'objectif sans dépassement	Nulle	$d_i^+ + d_i^-$

TAB. 2.4 – Différentes combinaisons de déviations.

Par exemple, si l'on veut approcher tous les objectifs par valeurs supérieures, on obtient le problème suivant :

$$
\left|\begin{array}{ll}
\text{minimiser} & \left(d_1^+, \cdots, d_k^+\right) \\
\text{avec} & f_1(\overrightarrow{x}) = F_1 + d_1^+ \\
 & \vdots \\
 & f_k(\overrightarrow{x}) = F_k + d_k^+ \\
 & \overrightarrow{h}(\overrightarrow{x}) = 0 \\
\text{et} & \overrightarrow{g}(\overrightarrow{x}) \leq 0
\end{array}\right.
$$

On voit que cette méthode permet de réduire le problème d'optimisation multiobjectif à la minimisation d'un vecteur. Pour minimiser ce vecteur, on peut procéder de plusieurs manières :

– On minimise la somme pondérée des déviations. Par exemple :

$$\min\left(d_1^+, d_2^-, d_3^+, d_4^+ + d_4^-\right)$$

se transformera de la manière suivante :

$$\min\left(2\cdot d_1^+ + 3\cdot d_2^- + 2\cdot d_3^+ + 1\cdot\left(d_4^+ + d_4^-\right)\right)$$

Les différents coefficients de pondération définissent un choix de l'utilisateur dans l'importance des fonctions objectif.

– On minimise séquentiellement les composantes du vecteur. On appelle cette méthode "optimisation lexicographique". Si l'on reprend l'exemple ci-dessus, on effectuera les opérations successives suivantes :
  - on minimise d_1^+,
  - on minimise d_2^- en gardant d_1^+ constant,
  - on minimise d_3^+ en gardant d_2^- et d_1^+ constants,
  - on minimise $d_4^+ + d_4^-$ en gardant d_3^+, d_2^- et d_1^+ constants.

2.7.3 Discussion

La méthode du but programmé, semblable à la méthode du but à atteindre, soulève les mêmes critiques. Dans certains cas, on peut "rater" des solutions cachées dans des concavités.

Pour plus d'informations, voir [Azarm 96], [Coello 98], [MOPGP] et [Miettinen 99].

2.8 L'ordonnancement lexicographique

2.8.1 Principe

Cette méthode est très intuitive. En effet, elle consiste à considérer les fonctions objectif les unes après les autres et à minimiser un problème d'optimisation monobjectif, en complétant au fur et à mesure l'ensemble des contraintes [Coello 98].

2.8.2 Présentation de la méthode

On part du problème P.

On procède ensuite en k étapes (autant d'étapes qu'il y a de fonctions objectif). Commençons avec la première fonction objectif. On résout :

$$\left|\begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right)\leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right)=0 \end{array}\right.$$

On note f_1^* la solution à ce problème.

Ensuite, on transforme la première fonction objectif en contrainte d'égalité puis on prend la seconde fonction objectif et on résout le problème suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_2\left(\overrightarrow{x}\right) \\ \text{avec} & f_1\left(\overrightarrow{x}\right)=f_1^* \\ & \overrightarrow{g}\left(\overrightarrow{x}\right)\leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right)=0 \end{array}\right.$$

On répète cette démarche jusqu'à la fonction objectif k. Alors, pour finir, on aura :

$$\left|\begin{array}{ll} \text{minimiser} & f_k(\overrightarrow{x}) \\ \text{avec} & f_1(\overrightarrow{x}) = f_1^*, \cdots, f_1(\overrightarrow{x}) = f_{k-1}^* \\ & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

La dernière valeur de $\overrightarrow{x}$ est celle qui minimise tous les objectifs.

2.8.3 Discussion

Cette méthode présente un inconvénient : elle requiert un choix de la séquence des objectifs à minimiser. Ce choix est largement arbitraire, un peu comme celui des coefficients de pondération, dans la méthode de pondération des fonctions objectif.

Deux ordonnancements différents de fonctions objectif n'aboutissent généralement pas à la même solution.

2.9 La méthode des contraintes d'égalité propres

2.9.1 Principe

Cette méthode (*Proper Equality Constraints* ou "contraintes d'égalité propres") a été présentée dans [Lin 76]. On l'appelle aussi la méthode des contraintes d'égalité strictes. Elle est très lourde d'un point de vue mathématique, mais ne nécessite aucune hypothèse de linéarité-convexité.

2.9.2 Présentation de la méthode

On part du problème P.

On le transforme de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & f_k(\overrightarrow{x}) \\ \text{avec} & f_1(\overrightarrow{x}) = \ _1 \\ & \vdots \\ & f_{k-1}(\overrightarrow{x}) = \ _{k-1} \\ & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

Jusqu'ici, la méthode est semblable à celle de la méthode du compromis. Nous allons, avant de continuer, présenter quelques notations.

- On note $\overrightarrow{\ } = \{\ _1, \cdots, \ _{k-1}\}$. C'est le vecteur des bornes des contraintes.
- On note X le domaine réalisable.
- On note $X\ = \{\overrightarrow{x} \in X \,|\, f_1(\overrightarrow{x}) = \ _1, \cdots, f_{k-1}(\overrightarrow{x}) = \ _{k-1}\}$.
- On note $D = \{\overrightarrow{\ } \in \mathbb{R}^{k-1} \,|\, X\ \neq \emptyset\}$. C'est l'ensemble des valeurs de $\overrightarrow{\ }$ telles que les contraintes additionnelles (obtenues après transformation du problème) ne soient pas trop restrictives et qu'il existe des valeurs dans X .
- On note $\Phi(\overrightarrow{\ }) = \inf\{f_k(\overrightarrow{x}) \,|\, \overrightarrow{x} \in X\ \}$. C'est la valeur minimale de la fonction objectif en prenant en compte toutes les contraintes.
- On note $B = \{\overrightarrow{\ } \in D \,|\, \Phi(\overrightarrow{\ }) < \infty$ et $f_k\left(\overrightarrow{\hat{x}}\right) = \Phi(\overrightarrow{\ }),\ \overrightarrow{\hat{x}} \in X\ \}$. B est une restriction de D.

Ces différentes définitions sont précisées à la figure 2.19.

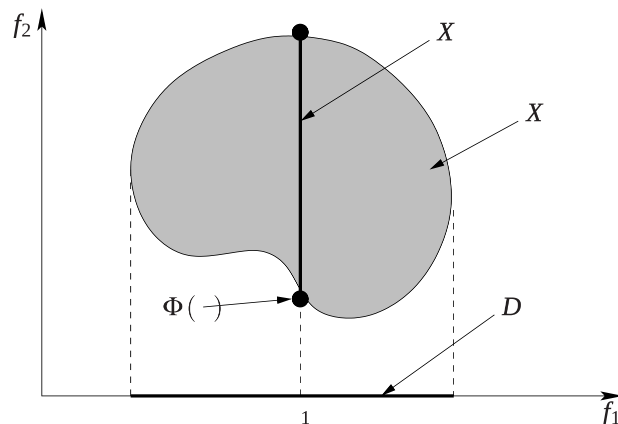


FIG. 2.19 – Représentation des diverses notions utilisées par la méthode, dans le cas $k = 2$.

Cette méthode comprend cinq étapes :

Etape 1 : Transformer tous les objectifs, sauf un, en contraintes d'égalité paramétriques. On obtient alors le problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & f_k\left(\overrightarrow{x}\right) \\ \text{avec} & f_1\left(\overrightarrow{x}\right) = \ _1, \cdots, f_{k-1}\left(\overrightarrow{x}\right) = \ _{k-1} \\ & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array} \right.$$

Etape 2 : Résoudre ce problème et déterminer $\Phi(\overrightarrow{\ })$, D et $\hat{x}\left(\overrightarrow{\ }\right)$. Voici la signification du dernier paramètre :

A chaque $\overrightarrow{\ } \in D$ correspond un $\overrightarrow{\hat{x}} = \overrightarrow{\hat{x}}\left(\overrightarrow{\ }\right)$ qui n'est pas nécessairement dans X ni dans X . Ce $\overrightarrow{\hat{x}}$ est tel que $f_k\left(\overrightarrow{\hat{x}}\left(\overrightarrow{\ }\right)\right) = \Phi(\overrightarrow{\ })$. On appelle cette valeur une solution "suprémum" du problème associé.

Etape 3 : Déterminer l'ensemble B et la solution optimale du problème associé $\overrightarrow{\hat{x}}\left(\overrightarrow{\ }\right)$ avec $\overrightarrow{\ } \in B$.

Etape 4 : Supprimer de B tous les vecteurs de contraintes $\overrightarrow{\ }$ qui ne soient pas des vecteurs propres (voir les deux propositions suivantes pour montrer qu'un vecteur n'est pas propre).

Etape 5 : Déterminer le reste final de B que l'on notera B^*.

Voici maintenant deux propositions qui nous serviront à vérifier qu'un vecteur est effectivement propre.

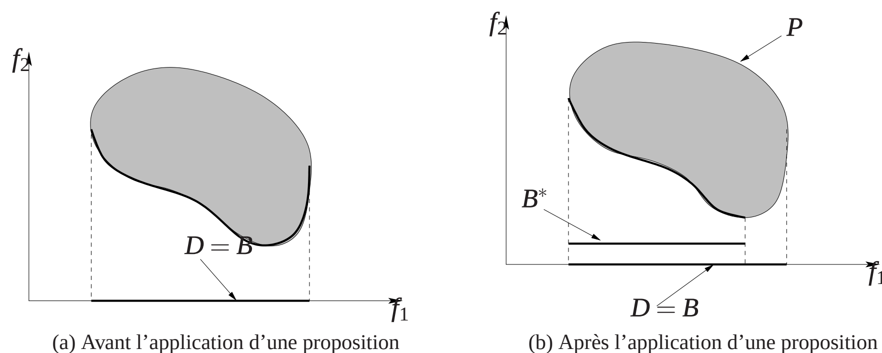
Proposition 1 :
Un vecteur $\overrightarrow{}_{0}$ est propre si l'une des assertions suivantes est vérifiée :

1. Φcroît sur D et $\Phi(\overrightarrow{}) \neq \Phi(\overrightarrow{}_{0})\ \forall \overrightarrow{} \in B$ tq $\overrightarrow{} \geq \overrightarrow{}_{0}$.
2. Φcroît sur D et est absolument croissante à droite de $\overrightarrow{}_{0}$ sur B.
3. Φcroît sur D et est absolument croissante sur B.
4. Φest absolument croissante à droite de $\overrightarrow{}_{0}$ sur D.
5. Φest absolument croissante sur le sous-ensemble $D\left(\overrightarrow{}_{0}\right) = \{\overrightarrow{} \in D | \overrightarrow{} \geq \overrightarrow{}\}$.
6. Φest absolument croissante sur D.
7. Φest absolument croissante sur $D\left(\overrightarrow{}_{0}\right)$ et est localement absolument croissante à droite de $\overrightarrow{}_{0}$.

Proposition 2 :
Un vecteur $\overrightarrow{}_{0} \in B$ n'est pas localement propre si $\frac{\partial^{+}\Phi(\overrightarrow{}_{0})}{\partial\ _{i}} < 0$ pour au moins un i tel que $1 \leq i \leq k-1$.

Voici, graphiquement (voir figure 2.20), à quoi servent ces propositions :

1. On part du fait que l'on a trouvé un ensemble B (figure 2.20-a).
2. On applique une des propositions pour filtrer l'ensemble B et supprimer ainsi les vecteurs qui ne sont pas propres (figure 2.20-b).
3. L'ensemble B^{*} contient maintenant la surface de compromis.



(a) Avant l'application d'une proposition

(b) Après l'application d'une proposition

FIG. 2.20 – Passage de B à B^{*} : suite à l'application d'une proposition.

Pour mieux comprendre cette méthode, nous allons traiter un exemple.

2.9.3 Un exemple concret

On reprend notre problème test “Schaffer's F2” :

$$\left|\begin{array}{ll} \text{minimiser} & f_{1}(x) = x^{2} \\ \text{minimiser} & f_{2}(x) = (x-2)^{2} \\ \text{avec} & x \in [0,2] \end{array}\right.$$

On choisit de conserver la seconde fonction (f_2) comme fonction objectif. On transforme l'autre fonction en contrainte d'égalité paramétrique. On applique donc l'étape 1. On a alors le problème suivant :

$$\begin{array}{|ll} \text{minimiser} & f_2(x) = (x-2)^2 \\ \text{avec} & f_1(x) = x^2 = \\ \text{et} & 0 \leq x \leq 2 \end{array}$$

On va maintenant déterminer D, $\Phi(\ )$ et $\hat{x}(\ )$. Ces déterminations correspondent à l'étape 2 de la méthode.

– Pour trouver D, on essaie de fusionner les deux contraintes en une seule. On remarque que $x = \sqrt{\ }$. On en déduit que $0 \leq \ \ \leq 4$. Donc $D = \{\ \ \in \mathbb{R} \,|\, 0 \leq \ \ \leq 4\}$.
– Pour trouver $\Phi(\ )$, on cherche une relation du type $f_2(x)$. On obtient facilement cette relation : $\Phi(\ ) = \left(\sqrt{\ } - 2\right)^2 = f_2(x)$ car $x = \sqrt{\ }$.
– Pour trouver $\hat{x}(\ )$, on cherche une relation du type $\hat{x} = f(\ )$ (où f est une fonction quelconque de la variable). On a cette relation facilement : $\hat{x}(\ ) = \sqrt{\ }$.

On passe maintenant à l'étape 3. On détermine l'ensemble $B = \{\ \ \in D \,|\, f_2(\hat{x}(\ )) = \Phi(\ )\}$; d'où $B = \{\ \ \in D \,|\, f_2(\hat{x}(\ )) = \left(\sqrt{\ } - 2\right)^2\}$.

On va maintenant réaliser les trois dernières étapes en un coup. Pour cela, on vérifie que $\frac{\partial \Phi(\ )}{\partial\ } < 0$. On calcule cette dérivée : $\frac{d\Phi(\ )}{d\ } = -\frac{1}{2\cdot\sqrt{\ }} < 0, \forall\ \ \in \mathbb{R}^{+*}$, donc on ne retire aucun point de B. On a ainsi, pour finir, $B^* = B$.

On va représenter graphiquement la surface de compromis. Pour cela, on choisit variant de 0 à 4 par pas de 1. On a le tableau suivant :

	$\hat{x}(\ )$	$\Phi(\ )$	$f_1(\hat{x}(\ ))$	$f_2(\hat{x}(\ ))$
0	0	4	0	4
1	1	1	1	1
2	$\sqrt{2}$	$\left(\sqrt{2}-2\right)^2$	2	$\left(\sqrt{2}-2\right)^2$
3	$\sqrt{3}$	$\left(\sqrt{3}-2\right)^2$	3	$\left(\sqrt{3}-2\right)^2$
4	2	0	4	0

On trouve alors le graphique de la figure 2.21.

2.9.4 Discussion

Cette méthode est une méthode littérale. Elle n'est pas facile à implémenter. A cette fin, il faut faire appel à deux algorithmes que l'on va présenter dans les sections 2.11 et 2.12.

2.10 La méthode des contraintes d'inégalité propres

2.10.1 Principe

Cette méthode (*Proper Inequality Constraints* ou "contraintes d'inégalité propres") est fortement inspirée de la méthode précédente et est présentée dans [Lin 77]. La différence réside dans le fait qu'elle transforme le problème de départ en utilisant des contraintes d'inégalité et non plus des contraintes d'égalité.

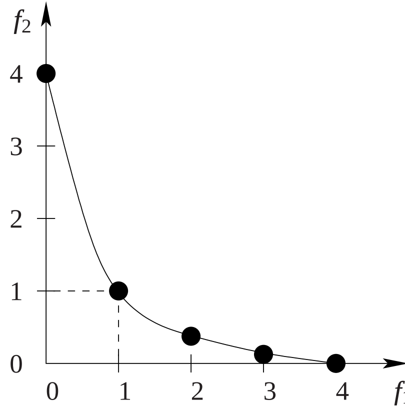


FIG. 2.21 – Problème test de Schaffer traité par la méthode des contraintes d'égalité propres.

2.10.2 Présentation de la méthode

On part du problème P.
On le modifie de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & f_k\left(\overrightarrow{x}\right) \\ \text{avec} & f_1\left(\overrightarrow{x}\right) \leq {}_{1} \\ & \quad \cdots \\ & f_{k-1}\left(\overrightarrow{x}\right) \leq {}_{k-1} \\ & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

Introduisons quelques notations.
On note $z_i = f_i\left(\overrightarrow{x}\right)$ et $\overrightarrow{\ } = \left({}_{1}, \cdots, {}_{k-1}\right)$.

- $Z = \left\{ \overrightarrow{f}\left(\overrightarrow{x}\right) \mid \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \text{ et } \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \right\}$: c'est l'ensemble des valeurs des fonctions objectif telles que les contraintes sur $\overrightarrow{x}$ soient respectées.
- $\overline{Z}$ est la frontière de Z.
- $Z\ = \left\{ \overrightarrow{z} \in \overline{Z} \mid z_i \leq {}_{i} \text{ avec } i \neq k \right\}$: c'est l'ensemble des valeurs des fonctions objectif telles qu'elles appartiennent à la frontière de Z et que $\overrightarrow{z} \leq \overrightarrow{\ }$.

Nous allons maintenant définir les ensembles A et P :

- $A = \left\{ \overrightarrow{\ } \in \mathbb{R}^{k-1} \mid \overline{Z\ } \text{ est non vide} \right\}$.
- $A\left(\overrightarrow{\ }_0\right) = \left\{ \overrightarrow{\ } \in A \mid \overrightarrow{\ } \geq \overrightarrow{\ }_0 \right\}$.
- $\ \left(\overrightarrow{\ }\right) = \inf\left\{ z_k \mid \overrightarrow{z} \in \overline{Z\ } \right\}$: c'est la valeur minimale de l'objectif $f_k\left(\overrightarrow{x}\right)$ quand les autres objectifs $\left(f_1\left(\overrightarrow{x}\right), \cdots, f_{k-1}\left(\overrightarrow{x}\right)\right)$ prennent leurs valeurs sur la frontière de $Z\ $.
- $P = \left\{ \left(\overrightarrow{\ }, \ \left(\overrightarrow{\ }\right)\right) \mid \overrightarrow{\ } \in A \right\}$.

Une fois que l'on a déterminé l'ensemble P, il ne reste qu'à déterminer l'ensemble D qui contient les solutions à notre problème. Pour cela, on prend les valeurs de l'ensemble P et on teste si ce sont des valeurs suprémum. A cette fin, on dispose d'un certain nombre de tests.

Théorème : quasi-suprémalité globale
$\left(\overrightarrow{\ }_0, \ \left(\overrightarrow{\ }_0\right)\right) \in D$ si et seulement si $\ \left(\overrightarrow{\ }\right) \neq \ \left(\overrightarrow{\ }_0\right)$ pour tout $\overrightarrow{\ } \in A$ tel que $\overrightarrow{\ } \geq \overrightarrow{\ }_0$

Théorème : quasi-suprémalité locale
$\left(\overrightarrow{\ }_0, \ \left(\overrightarrow{\ }_0\right)\right) \in D$ si et seulement si, pour un voisinage $N\left(\overrightarrow{\ }_0, \Delta\right)$, $\ \left(\overrightarrow{\ }\right) \neq \ \left(\overrightarrow{\ }_0\right)$ pour tout $\overrightarrow{\ } \in N\left(\overrightarrow{\ }_0, \Delta\right) \bigcap A\left(\overrightarrow{\ }_0\right)$ tel que $\overrightarrow{\ } \neq \overrightarrow{\ }_0$.

2.10.3 Discussion

Cette méthode a été très utilisée et a donné naissance à quelques algorithmes que nous étudierons dans les sections 2.11 et 2.12.

2.11 L'algorithme de Lin-Tabak

2.11.1 Principe

Cet algorithme [Tabak 79] est basé sur la méthode précédente.

2.11.2 Présentation de la méthode

On part du problème P.
On transforme le problème de la manière suivante :

$$
\left|
\begin{array}{ll}
\text{minimiser} & f_k\left(\overrightarrow{x}\right) \\
\text{avec} & f_1\left(\overrightarrow{x}\right) \leq \ _1, \cdots, f_{k-1}\left(\overrightarrow{x}\right) \leq \ _{k-1} \\
 & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\
\text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0
\end{array}
\right.
$$

On note $\Phi(\overrightarrow{\ }) = f_k\left(\overrightarrow{x_0}\left(\overrightarrow{\ }\right)\right)$, où $\overrightarrow{\ } = [\ _1, \cdots, \ _{k-1}]^t$.
L'algorithme se décompose en cinq étapes :

Etape 1 : Résoudre le problème avec une valeur initiale $\overrightarrow{\ ^{(0)}}$, ce qui donne un optimum global $\Phi\left(\overrightarrow{\ ^{(0)}}\right) = \Phi^{(0)}$.

Etape 2 : Modifier $\ _1^{(0)} \rightarrow \ _1^{(0)} + \Delta \ _1 = \ _1^{(1)}$ avec $\Delta \ _1 > 0$ et recommencer la résolution du problème. On obtient alors un optimum global $\Phi^{(1)}$.

Étape 3 : Effectuer les comparaisons suivantes :

1. Si $\Phi^{(1)} = \Phi^{(0)}$ alors il faut détruire $\ _1^{(0)}$ car il n'est pas propre.
2. Si $\Phi^{(1)} \neq \Phi^{(0)}$ alors accepter $\ _1^{(0)}$ et $\ _1^{(1)}$. En fait, on doit avoir $\Phi^{(1)} < \Phi^{(0)}$. Si l'inégalité est inversée, la solution du problème précédent était juste une solution locale.

Etape 4 : Répéter les étapes 2 et 3 avec des valeurs croissantes de $\ _1$ jusqu'à ce qu'une frontière de la région propre dans la direction de $\ _1$ soit atteinte. Cela deviendra évident avec l'aboutissement de l'étape 3-1.

Etape 5 : Répéter les étapes 2 à 4 pour un autre $\ _i$, $i = \{2, \cdots, k-1\}$.

2.12 L'algorithme de Lin-Giesy

2.12.1 Principe

Cette méthode [Giesy 78] repose également sur la méthode des contraintes d'inégalité propres. Il existe, cependant, un théorème qui garantit, dans certains cas, la convergence de l'algorithme.

2.12.2 Présentation de la méthode

On introduit d'abord quelques notations :

- On note $\overrightarrow{\ ^{(0)}}$ le choix de la valeur initiale de $\overrightarrow{\ }$ appelée "seuil de satisfaction". Cette valeur doit être donnée par l'utilisateur. Dans cet algorithme, le vecteur $\overrightarrow{\ }$ est de dimension k.
- On note $\overrightarrow{\ ^{(i)}}$ la valeur du vecteur $\overrightarrow{\ }$ atteinte à l'itération i ($i = \{1, \cdots, k\}$).
  Ici, k prend la même valeur que la variable k ci-dessus.
- On note une permutation de l'ensemble $\{1, 2, \cdots, k\}$.
  Par exemple, si $A = \{1, 2, 3\}$ alors on peut prendre $= \{2, 1, 3\}$ et $_1 = 2$, $_2 = 1$ et $_3 = 3$.
- A l'itération i, la description de l'algorithme est la suivante :

  Etape i_1 : Résoudre le problème

$$\left|\begin{array}{ll} \text{minimiser} & f_{\ i}(\overrightarrow{x}) \\ \text{avec} & f_j(\overrightarrow{x}) \leq \ _j,\ j = \{1, \cdots, k\},\ j \neq \ _i \\ & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

  On note la solution $\overrightarrow{x^{(i)}}$.

  Pour l'itération 1 seulement, s'il n'y a pas de solution ou si $f_{\ 1}\left(\overrightarrow{x^{(1)}}\right) \geq \ _1^{(0)}$, alors un choix moins restrictif de $\overrightarrow{\ ^{(0)}}$ doit être fait.

  Etape i_2 : On pose $\overrightarrow{\ ^{(i)}} = \overrightarrow{f}\left(\overrightarrow{x^{(i)}}\right)$.

On admettra le théorème suivant, concernant la convergence de la méthode.

Théorème : convergence
Si l'algorithme trouve une solution lors de l'itération 1, alors il aboutira à toutes les k itérations et $\overrightarrow{x^{(k)}}$ est la solution optimale au sens de Pareto et $\overrightarrow{f}\left(\overrightarrow{x^{(k)}}\right) \leq \overrightarrow{\ ^{(0)}}$.

2.13 Bibliographie commentée

[Azarm 96] Site Internet présentant les méthodes multiobjectifs les plus courantes comme la méthode de pondération des fonctions objectif par exemple. La présentation de ces méthodes est très orientée "utilisateur". Peu de détails sont donnés sur

l'aspect mathématique de ces méthodes mais, en revanche, toute la démarche expliquant l'utilisation de ces méthodes pour approcher la surface de compromis est détaillée.

[Ehrgott 00] Un ouvrage présentant les méthodes d'optimisation multiobjectif. Le niveau mathématique de l'exposé est plutôt élevé. C'est un ouvrage de référence pouvant servir de base de données des propriétés mathématiques des méthodes d'optimisation multiobjectif.

Chapitre 3

Les méthodes interactives

3.1 Introduction

Les méthodes interactives permettent de chercher une et une seule solution. Elles forment la famille des méthodes progressives et permettent à l'utilisateur de déterminer ses préférences vis-à-vis d'un compromis entre objectifs au cours de l'optimisation. Ces méthodes sont à comparer aux méthodes :

– à préférence *a priori*, où l'utilisateur choisit le compromis à réaliser entre objectifs avant de lancer l'optimisation ;
– à préférence *a posteriori*, où l'utilisateur n'a pas à choisir de compromis avant de lancer l'optimisation. La méthode d'optimisation calcule toutes les solutions efficaces et l'utilisateur peut ainsi les comparer et en choisir une.

Nous allons maintenant présenter quelques méthodes interactives (ou plutôt quelques principes d'interaction).

3.2 La méthode du compromis par substitution

3.2.1 Principe

Cette méthode (*Surrogate Worth Tradeoff* (SWT) ou "méthode du compromis par substitution") a été très utilisée pour l'optimisation de ressource en eau [Haimes et al. 75]. Elle est basée sur la méthode du compromis, à laquelle on a ajouté un processus interactif, pour que la méthode converge vers la solution la plus susceptible de satisfaire l'utilisateur.

3.2.2 Présentation de la méthode

On part du problème P.

Cette méthode se décompose en sept étapes :

Etape 1 : Trouver le minimum de la fonction f_j en résolvant :

$$\left|\begin{array}{ll} \text{minimiser} & f_j(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

La solution à ce problème devient la j^e composante du vecteur $\overrightarrow{f}_{min}$, qui regroupe les $k-1$ valeurs minimales de f_j, $j = \{2, \cdots, k\}$. Si possible, on peut chercher à cette étape les composantes du vecteur $\overrightarrow{f}_{max}$, qui regroupe les $k-1$ valeurs maximales de f_j, $j = \{2, \cdots, k\}$.

Etape 2 : Choisir la valeur de départ de $_j \geq f_{j_{min}}$, $j = \{2, \cdots, k\}$.

Etape 3 : Résoudre le problème suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{avec} & f_2(\overrightarrow{x}) \leq {}_2, \cdots, f_k(\overrightarrow{x}) \leq {}_k \\ & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

Si certaines des contraintes de ce problème ne peuvent pas être respectées, on pose $_j = f_j(\overrightarrow{x})$ pour j correspondant aux index des contraintes non respectées (on relâche les contraintes). Il faut ensuite recommencer cette étape. Si les contraintes sont respectées, on appelle $\overrightarrow{x}^*$ le vecteur solution et on passe à l'étape suivante.

Etape 4 : Si les contraintes sont maintenant suffisamment relâchées, on passe à l'étape 5 sinon, on choisit une nouvelle valeur de $_j \geq f_{j_{min}}$, pour tout $j = \{2, \cdots, k\}$ et on retourne à l'étape 3.

Une méthode pour sélectionner une nouvelle valeur de consiste à commencer par une grande valeur de puis à diminuer chaque $_j$, pour tout $j = \{2, \cdots, k\}$ d'un certain $\Delta_j > 0$, chaque fois que les contraintes sont respectées.

Si les contraintes ne sont pas respectées, on pose $_j = f_j(\overrightarrow{x})$ avec j index des contraintes non respectées (ici $\overrightarrow{x}$ désigne la valeur courante de la solution).

Etape 5 : On demande maintenant à l'utilisateur de choisir les coefficients W_{1j}, pour tout $j = \{2, \cdots, k\}$. Ces coefficients représentent l'avis de l'utilisateur vis-à-vis d'un accroissement de la fonction f_j de $_{1j}$ unités (le mode de calcul de $_{1j}$ est précisé plus loin). Ce coût est mesuré sur une échelle de -10 à +10, où -10 représente un avis défavorable de l'utilisateur vis-à-vis de cet accroissement et +10 un avis favorable. 0 représente l'indifférence.

Cette opération est répétée pour j variant de 2 à k.

Etape 6 : On répète l'étape 5 jusqu'à ce que l'on trouve pour les coefficients W_{1j}, $j = \{2, \cdots, k\}$ des valeurs égales à zéro. On note f_j^*, pour tout $j = \{2, \cdots, k\}$ les valeurs des fonctions objectif correspondant à ces coefficients.

Etape 7 : Le vecteur solution préféré $\overrightarrow{x}^*$ est déterminé en résolvant le problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{avec} & f_2(\overrightarrow{x}) = f_2^*, \cdots, f_k(\overrightarrow{x}) = f_k^* \\ & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array} \right.$$

Le plus difficile dans cette méthode est de bien comprendre la signification du coefficient W_{1j}. Pour cela, voici une procédure que l'on peut suivre pour évaluer W_{1j} :

Un point possible pour la résolution du problème est le suivant :
- $f_2(\overrightarrow{x})$ [unités de f_2],
- $f_3(\overrightarrow{x})$ [unités de f_3] et
- $f_1(\overrightarrow{x})$ [unités de f_1].

On se pose alors les questions suivantes :

Pour W_{12} : A ce point de fonctionnement, seriez-vous prêt à augmenter f_2 de $_{12}$ [unités de f_2] pour une réduction de f_1 de 1 [unité de f_1] ? Donnez votre réponse sur une échelle de -10 (pas d'accord) à +10 (d'accord).

Pour W_{13} : A ce point de fonctionnement, seriez-vous prêt à augmenter f_3 de $_{13}$ [unités de f_3] pour une réduction de f_1 de 1 [unité de f_1] ? Donnez votre réponse sur une échelle de -10 (pas d'accord) à +10 (d'accord).

Cette méthode va permettre, dans un premier temps, de trouver une zone de satisfaction (entre deux 0 de W_{12} ou de W_{13}, ou entre un 0 de W_{12} et un 0 de W_{13}. Voir les figures 3.1 et 3.2). Une fois que l'on a trouvé cette zone de satisfaction, on peut résoudre le problème posé en utilisant la méthode du compromis, en se restreignant aux valeurs de la zone de satisfaction.

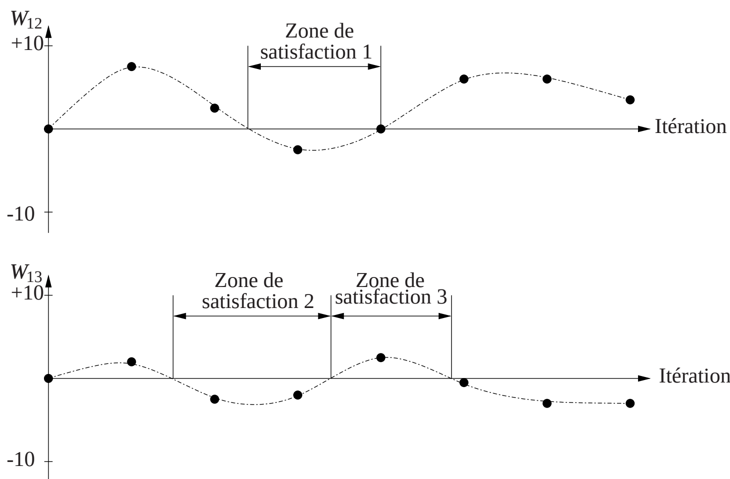


FIG. 3.1 – Les zones de satisfaction dans la méthode SWT - Exemple 1.

La valeur du coefficient $_{ij}$ se calcule de la manière suivante :

$$_{ij} = -\frac{\partial f_i}{\partial f_j} \tag{3.1}$$

Dans notre cas, on ne calcule que :

$$_{1j} = -\frac{\partial f_1}{\partial f_j} \tag{3.2}$$

car on fait toujours notre comparaison en référence à la fonction objectif f_1.

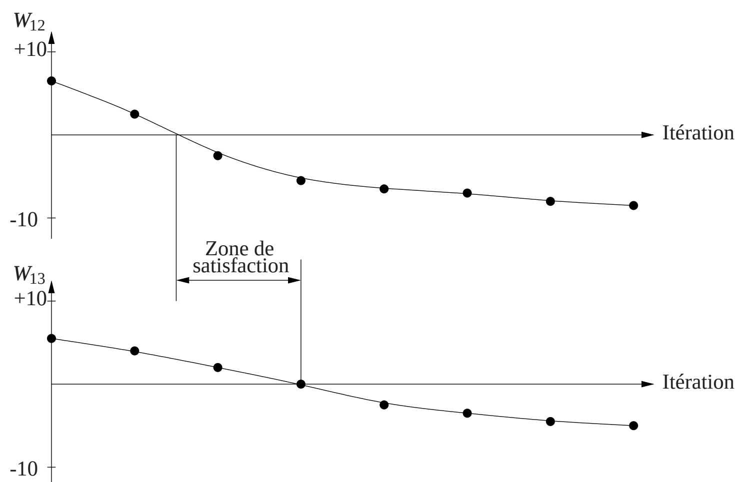


FIG. 3.2 – Les zones de satisfaction dans la méthode SWT - Exemple 2.

3.2.3 Discussion

Dans cette méthode, il n'est pas toujours évident de savoir quel prix on est prêt à payer un accroissement de la valeur d'une fonction. Dans la plupart des cas, on peut être tenté de répondre un peu au hasard. Ce type de réponse peut alors empêcher la méthode de trouver le ou les bons optima.

De plus, cette méthode n'est applicable qu'à des problèmes dont la fonction objectif de référence est dérivable (à cause du coefficient $_{1j}$).

3.3 La méthode de Fandel

3.3.1 Principe

Le but de cette méthode est d'aider l'utilisateur dans le choix de ses coefficients de pondération [Eschenauer et al. 90].

3.3.2 Présentation de la méthode

On part du problème P.

On modifie le problème de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & \oplus_{i=1}^{k} w_i \cdot f_i(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

On a aussi $w_i \geq 0$ et $\oplus_{i=1}^{k} w_i = 1$. On retrouve cette condition dans la méthode de pondération des fonctions objectif (voir section 2.1.2).

Comme pour la méthode de pondération des fonctions objectif, la variation des coefficients w_i permet de déterminer les points de la surface de compromis. En revanche, dans la méthode de Fandel, on suppose que ces coefficients ne sont pas connus. On suppose aussi

que l'utilisateur cherche une solution qui soit proche de la solution idéale (ce terme est défini plus loin).

Pour obtenir ces coefficients, on procède comme suit :

– Dans un premier temps, on calcule le minimum de chaque fonction objectif (que l'on notera $\overline{f}_j$) en respectant les contraintes. Cela revient à résoudre le problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & f_j\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array} \right.$$

avec $j = \{1, \cdots, k\}$.
On note $\overrightarrow{x}_j^*$ la solution de ce problème.
On note $\overrightarrow{f}^{*j} = \overrightarrow{f}\left(\overrightarrow{x_j^*}\right)$ la valeur du vecteur de fonctions objectif correspondant à cette solution.
Pour $j = 4$, ce vecteur aura pour composantes :

$$\overrightarrow{f}^{*4} = \left[f_1\left(\overrightarrow{x_4^*}\right), \cdots, f_3\left(\overrightarrow{x_4^*}\right), \overline{f_4}, f_5\left(\overrightarrow{x_4^*}\right), \cdots, f_k\left(\overrightarrow{x_4^*}\right)\right]$$

– On forme ensuite le vecteur objectif idéal :

$$\overrightarrow{\overline{f}} = \left(\overline{f}_1, \cdots, \overline{f}_k\right)^t \tag{3.3}$$

On peut aussi former la matrice B :

$$B = \left(\overrightarrow{f}^{*1}, \cdots, \overrightarrow{f}^{*k}\right)^t \tag{3.4}$$

Cette matrice contient les valeurs $\overline{f}_j$ sur sa diagonale.
Pour un problème à deux objectifs, on a le schéma représenté figure 3.3.

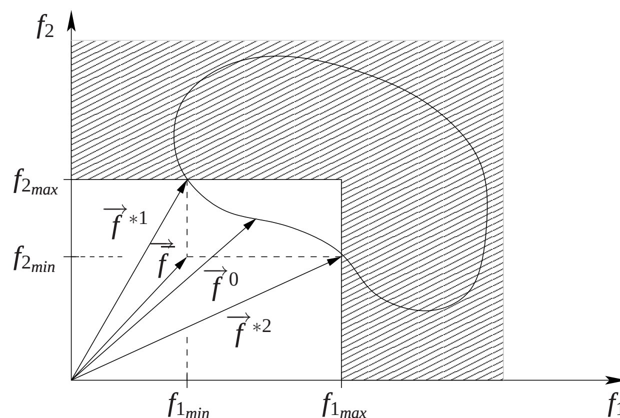


FIG. 3.3 – La méthode de Fandel (étape 1).

Le vecteur $\overrightarrow{\overline{f}}$ représente donc un objectif "idéal" pour l'utilisateur mais qu'on ne peut pas atteindre en réalité. Ce vecteur est le but que se fixe l'utilisateur (il est prêt à tout faire pour atteindre cet objectif).

Reportez-vous à la figure 3.3 pour une illustration des différentes notations introduites jusqu'ici. Sur cette figure, le domaine grisé représente la partie du domaine de recherche qui n'est pas accessible à cause des contraintes. Les variables $f_{1_{min}}$, $f_{1_{max}}$, $f_{2_{min}}$ et $f_{2_{max}}$ délimitent les domaines de variation des fonctions objectif f_1 et f_2 dans la zone réalisable.

A l'étape M, l'utilisateur va obtenir une solution qui sera proche de la solution idéale. Ici, on calcule un vecteur $\overrightarrow{f}^M$ qui servira à réduire la taille de l'espace de recherche. Ce vecteur s'écrit de la manière suivante :

$$\overrightarrow{f}^M = \frac{1}{k} \cdot \bigoplus_{i=1}^{k} \overrightarrow{f}^{*i} \tag{3.5}$$

– On va maintenant réduire la taille de l'espace de recherche en jouant sur les contraintes. On appelle notre nouvel espace de contraintes $\hat{Y}^M$:

$$\hat{Y}^M = \left\{ \overrightarrow{f}(\overrightarrow{x}) \in \mathbb{R}^k \,|\, \overrightarrow{g}(\overrightarrow{x}) \leq 0,\ \overrightarrow{h}(\overrightarrow{x}) = 0 \text{ et } \overrightarrow{f}(\overrightarrow{x}) \leq \overrightarrow{f}^M \right\} \tag{3.6}$$

– Dans ce nouvel espace de contraintes, on minimise nos fonctions objectif :

$$\left| \begin{array}{ll} \text{minimiser} & f_j(\overrightarrow{x}),\ j = \{1, \cdots, k\} \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ & \overrightarrow{h}(\overrightarrow{x}) = 0 \\ \text{et} & \overrightarrow{f}(\overrightarrow{x}) \leq \overrightarrow{f}^M \end{array} \right.$$

On note $\hat{f}^{*j}$ la solution de ce problème et $\overrightarrow{\hat{x}}^{*j}$ le vecteur décision correspondant.

– Comme précédemment, on forme la matrice $\hat{B}$:

$$\hat{B} = \left(\overrightarrow{\hat{f}}^{*1}, \cdots, \overrightarrow{\hat{f}}^{*k} \right)^t \tag{3.7}$$

où $\overrightarrow{\hat{f}}^{*k}$ désigne le vecteur $\overrightarrow{f}\left(\overrightarrow{\hat{x}}^{*k}\right)$.

Cette matrice $\hat{B}$ définit un hyperplan (une droite dans le cas d'un problème à deux fonctions objectif) qui a pour équation :

$$\hat{B} \cdot \overrightarrow{a} = c \cdot \overrightarrow{e} \tag{3.8}$$

$$\begin{array}{ll} \text{avec} & \overrightarrow{a} = (a_1, \cdots, a_k)^t, \\ & \overrightarrow{a} > 0,\ \bigoplus_{i=1}^{k} a_j = 1, \\ & c = \text{constante}, \\ & \overrightarrow{e} = (1, \cdots, 1)^t \end{array}$$

Le vecteur $\overrightarrow{a}$ et la constante c sont les inconnues du système d'équations. Le but va être de trouver une valeur de c telle que l'hyperplan défini par l'équation précédente vienne tangenter l'espace des valeurs des fonctions objectif. Le produit $\hat{B} \cdot \overrightarrow{a}$ avec les contraintes pré citées correspond à une combinaison convexe des vecteurs $\overrightarrow{\hat{f}}^{*i}$. Ce produit donne donc bien un hyperplan passant par les extrémités des vecteurs $\overrightarrow{\hat{f}}^{*i}$.

Le fonctionnement de cette étape est représenté à la figure 3.4. Sur cette figure, on peut voir les deux vecteurs $\overrightarrow{\hat{f}}^{*1}$ et $\overrightarrow{\hat{f}}^{*2}$ qui correspondent aux vecteurs calculés lors de la résolution du problème f^*. Le vecteur $\overrightarrow{f}^M$ correspond aux bornes du domaine réalisable. Ces bornes sont données par l'utilisateur.

Exemple 1 :

Une solution du problème précédent est la suivante :
$\overrightarrow{\hat{f}}^{*1} = (1,0)$
$\overrightarrow{\hat{f}}^{*2} = (0,2)$
$\hat{B} \cdot \overrightarrow{a} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ 1 - a_1 \end{pmatrix} = \begin{pmatrix} c \\ c \end{pmatrix}$
on a donc :
$2 - 2 \cdot a_1 = c = a_1$
$a_1 = \frac{2}{3}$

Exemple 2 :

Une solution du problème précédent est la suivante :
$\overrightarrow{\hat{f}}^{*1} = (2,3)$
$\overrightarrow{\hat{f}}^{*2} = (4,1)$
$\hat{B} \cdot \overrightarrow{a} = \begin{pmatrix} 2 & 4 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ 1 - a_1 \end{pmatrix} = \begin{pmatrix} c \\ c \end{pmatrix}$
on a donc :
$2 \cdot a_1 + 4 - 4 \cdot a_1 = c = 3 \cdot a_1 + 1 - a_1$
$-4 \cdot a_1 + 3 = 0$
$a_1 = \frac{3}{4}$

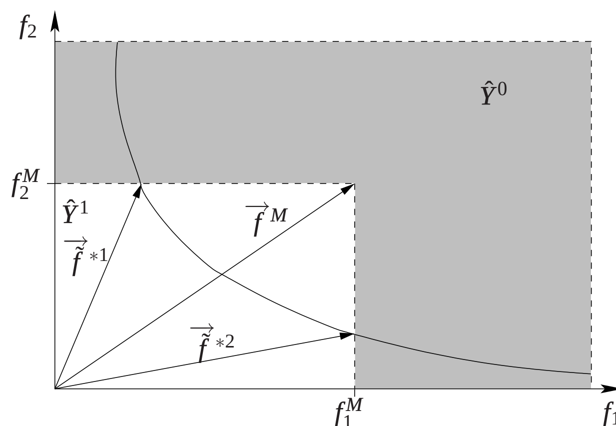


FIG. 3.4 – La méthode de Fandel (étape 2).

De manière à déterminer les coefficients de pondération d'une solution de compromis désirée à l'intérieur du sous-espace $\hat{Y}^{\mathcal{M}}$, l'hyperplan (une droite si l'on considère un espace de dimension 2) est déplacé en direction de la solution optimale au sens de Pareto, en minimisant la constante c jusqu'à ce que l'hyperplan soit tangent en un point à la surface de compromis. Ce processus est représenté graphiquement à la figure 3.5. Sur cette figure, après avoir calculé les vecteurs $\overrightarrow{\hat{f}}^{*1}$ et $\overrightarrow{\hat{f}}^{*2}$, on détermine l'équation de l'hyperplan (B) puis on cherche un hyperplan parallèle qui vient tangenter le domaine réalisable. Le point de tangence définit alors une nouvelle solution notée $\overrightarrow{\tilde{f}}$.

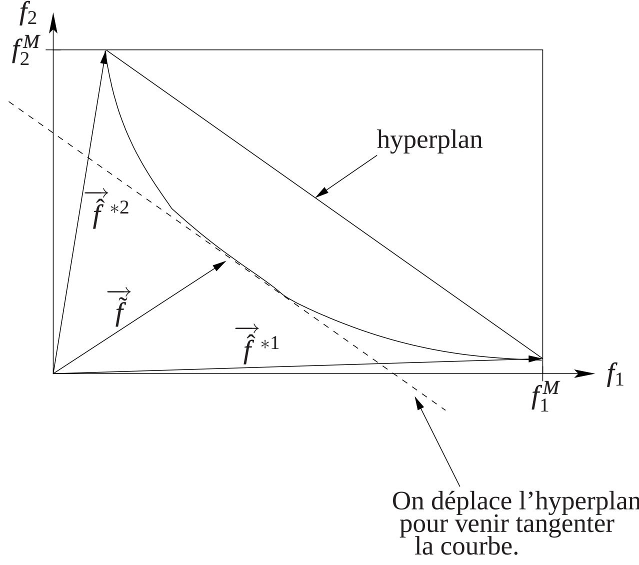


FIG. 3.5 – La méthode de Fandel (étape 3).

Ce problème d'optimisation linéaire nous donne un vecteur $\overrightarrow{\tilde{f}}$, qui correspond au vecteur des coefficients de pondération $\overrightarrow{w}$.

La solution de notre problème de substitution peut maintenant être déterminée.

Soit l'utilisateur accepte cette solution, soit il commence une nouvelle itération, en précisant le domaine de recherche. Ceci peut se faire en fixant un certain nombre de bornes maximal sur un certain nombre de fonctions objectif (m fonctions objectif par exemple). Notons $\overline{Y}_i$ les bornes supérieures des m fonctions objectif sélectionnées pour restreindre l'espace de recherche. Les minima individuels du nouveau sous-espace $\hat{Y}$ sont déterminés par :

$$\left|\begin{array}{ll} \text{minimiser} & f_j(\overrightarrow{x}),\ j=\{1,\cdots,k\} \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ & \overrightarrow{h}(\overrightarrow{x}) = 0 \\ \text{et} & f_i(\overrightarrow{x}) \leq \overline{Y}_i \end{array}\right.$$

l'index i désignant toutes les fonctions objectif sélectionnées pour restreindre l'espace de recherche.

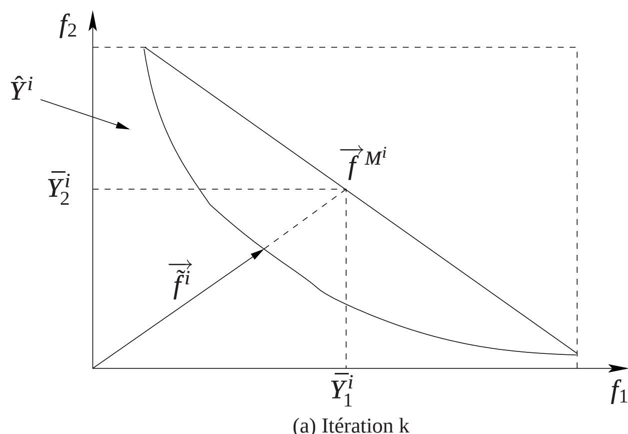
On a $\hat{Y}^{i+1} = \left\{\overrightarrow{f}(\overrightarrow{x}) \in \mathbb{R}^k \mid \overrightarrow{g}(\overrightarrow{x}) \leq 0,\ \overrightarrow{h}(\overrightarrow{x}) = 0,\ f_j(\overrightarrow{x}) \leq \overline{Y}^j\right\}$.

L'index j désigne toutes les fonctions objectif sélectionnées pour restreindre l'espace de recherche.

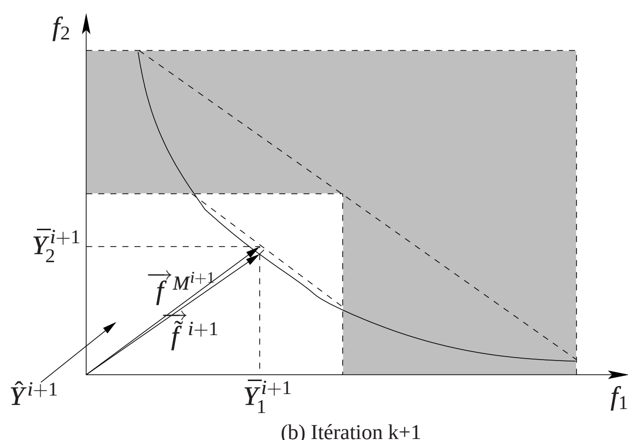
L'itération commence par le calcul d'un nouveau point central $\overrightarrow{f}^{M^{k+1}}$ (voir figure 3.6).

A la figure 3.6-a, on peut voir l'itération k de la méthode de Fandel. Un espace de recherche $\hat{Y}^k$ a été déterminé. On construit le vecteur médian $\overrightarrow{f^{Mk}}$ en utilisant la formule 3.5, puis on fait une recherche de la solution $\overrightarrow{\tilde{f}}^k$ dans la direction du vecteur médian.

A la figure 3.6-b, on peut voir l'itération $k+1$ de la méthode de Fandel. Après avoir calculé le vecteur médian $\overrightarrow{f^{Mk}}$, on définit un nouvel espace de recherche en utilisant ce vecteur médian. On calcule ensuite un nouveau vecteur médian $\overrightarrow{f^{Mk+1}}$ qui va définir un nouvel espace



(a) Itération k



(b) Itération k+1

FIG. 3.6 – La méthode de Fandel (étape 4).

de recherche $\hat{Y}^{k+1}$. On fait ensuite une recherche dans la direction du vecteur médian qui donnera le vecteur $\overrightarrow{\tilde{f}}^{k+1}$.

On voit donc, au fur et à mesure que l'on répète la méthode, que l'on converge vers un point qui, normalement, doit satisfaire l'utilisateur.

3.3.3 Discussion

Le problème de cette méthode, commun à toutes les méthodes interactives, est que l'on ne peut trouver qu'un seul point solution, et non la totalité de la surface de compromis. On voit aussi que cette méthode fait l'hypothèse que le vecteur solution qui satisfera l'utilisateur est le vecteur le plus proche de la solution idéale. C'est une hypothèse forte qui fait que cette méthode peut ne pas convenir à tous les utilisateurs. Pour finir, le mode d'interaction (réduction de l'espace de recherche par action sur des bornes de contraintes) fait que, pour une bonne utilisation de la méthode, l'utilisateur doit avoir une bonne connaissance *a priori* du problème. Il doit être capable de trouver de bonnes bornes pour restreindre son problème.

3.4 La méthode STEP

3.4.1 Principe

Cette méthode est similaire à la méthode de Fandel. Ici aussi, les informations sur la préférence de l'utilisateur permettent de restreindre l'espace de recherche étape par étape [Eschenauer et al. 90].

3.4.2 Présentation de la méthode

On part du problème P.

On transforme ce problème de la manière suivante :

$$
\left|\begin{array}{ll}
\text{minimiser} & \\
\text{avec} & \left[\overrightarrow{f}(\overrightarrow{x}) - \overrightarrow{\overline{Y}}\right]^t \cdot \overrightarrow{w} < \\
 & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\
 & \overrightarrow{h}(\overrightarrow{x}) = 0 \\
\text{et} & \overrightarrow{f}(\overrightarrow{x}) \leq \overrightarrow{\overline{Y}}
\end{array}\right.
$$

où le vecteur $\overrightarrow{\overline{Y}}$ correspond à un vecteur de bornes supérieures servant à restreindre l'espace de recherche. Le vecteur $\overrightarrow{w}$ est défini dans la suite de cette section.

De la même manière que l'approche précédente, on forme la matrice B.

Les coefficients de pondération w_j des différentes fonctions objectif f_j sont nécessaires pour la résolution du problème. On va déterminer ces coefficients. On va donner un poids plus important aux fonctions f_j dont l'excursion est importante (f_j prend ses valeurs dans un domaine étendu). Ces coefficients ont la forme suivante :

$$w_j = \frac{v_j}{\bigoplus_{i=1}^{k} v_i} \tag{3.9}$$

et :

$$v_j = \frac{\max_i \left\{f_j^{*i}\right\} - f_j^{*j}}{\min_i \left\{f_j^{*i}\right\}} \cdot \frac{1}{f_j^{*j}} \tag{3.10}$$

avec $i = 1, \cdots, k$ et f_j^{*i} a la même signification qu'à la page 78.

De cette manière, les coefficients respectent $\overrightarrow{w} > 0$ et $\bigoplus_{j=1}^{k} w_j = 1$.

Analysons maintenant le comportement des équations donnant les v_j et les w_j à travers un exemple.

Exemple

A l'itération 10, on obtient les valeurs des cinq fonctions objectif suivantes :

f^{*1}	$\{1,4,3,6,2\}$
f^{*2}	$\{4,1,4,3,4\}$
f^{*3}	$\{3,2,1,2,3\}$
f^{*4}	$\{6,3,6,2,3\}$
f^{*5}	$\{7,5,3,2,1\}$

On a alors les résultats suivants :

v_1	$v_1 = \frac{7-1}{1} \cdot \frac{1}{1} = 6$	w_1	$\frac{6}{19} = 0.315$
v_2	$v_2 = \frac{5-1}{1} \cdot \frac{1}{1} = 4$	w_2	$\frac{4}{19} = 0.210$
v_3	$v_3 = \frac{6-1}{1} \cdot \frac{1}{1} = 5$	w_3	$\frac{5}{19} = 0.263$
v_4	$v_4 = \frac{6-2}{2} \cdot \frac{1}{2} = 1$	w_4	$\frac{1}{19} = 0.052$
v_5	$v_5 = \frac{4-1}{1} \cdot \frac{1}{1} = 3$	w_5	$\frac{3}{19} = 0.158$

Comme on peut le voir sur ces résultats, la méthode S.T.E.P. attribue des poids plus importants aux fonctions objectif qui présentent une différence entre le minimum de la fonction f^{*i} et le maximum de la fonction f^{*i} la plus importante. Dans cet exemple, on a ce cas de figure sur la fonction objectif f^{*1}.

On pourra aussi remarquer que ce mode de calcul des poids ne fonctionne pas avec des fonctions objectif qui peuvent être nulles.

On constate que, comme avec la méthode de Fandel, les coefficients de pondération ont une importance moindre, car l'interaction avec l'utilisateur se fait par le choix des bornes supérieures $\overrightarrow{\overline{Y}}$ et non par le choix des coefficients. De plus, la méthode attribue des coefficients plus petits aux fonctions objectif dont les valeurs sont les plus grandes.

Une fois les coefficients déterminés, on calcule une première solution $\overrightarrow{\widetilde{f}}^{\,0}$ de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & \overrightarrow{f}(\overrightarrow{x})^t \cdot \overrightarrow{w} \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

L'utilisateur peut alors comparer la solution $\overrightarrow{\widetilde{f}}^{\,0}$ avec la solution idéale. Si certaines composantes du vecteur des fonctions objectif ne sont pas suffisamment satisfaites, l'utilisateur peut relâcher la borne supérieure de la contrainte correspondante de la manière suivante :

$$\overline{Y_j} = \widetilde{f_j} + \Delta f_j \tag{3.11}$$

avec $j \in \{1, \cdots, k\}$.

Sinon, la solution préférée de l'utilisateur est donnée par $\overrightarrow{\widetilde{f}}$.

On a alors obtenu une nouvelle solution de compromis. De plus, le coefficient de pondération w_j de la j^e fonction objectif est mis à zéro. Le problème a alors la forme suivante :

$$\left|\begin{array}{ll} \text{minimiser} & \left[\overrightarrow{f}(\overrightarrow{x}) - \overrightarrow{\overline{Y}}\right]^t \cdot \overrightarrow{w} \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ & \overrightarrow{h}(\overrightarrow{x}) = 0 \\ \text{et} & \overrightarrow{f}(\overrightarrow{x}) \leq \overrightarrow{\overline{Y}} \end{array}\right.$$

avec $w_j = 0$ et $j \in \{1, \cdots, k\}$.

Il est donc restreint au sous-espace de recherche $\hat{Y}$ et possède la plus petite déviation par rapport au vecteur des fonctions objectif idéal $\overrightarrow{\overline{f}}$.

Dans le cas d'un problème à deux dimensions, l'extrémité du vecteur $\overrightarrow{\tilde{f}}$ est située à l'intersection de la frontière du domaine $\hat{Y}$ et de la droite de pente $\frac{w_2}{w_1}$ passant par le point idéal $\overrightarrow{\overline{f}}$ (voir figure 3.7).

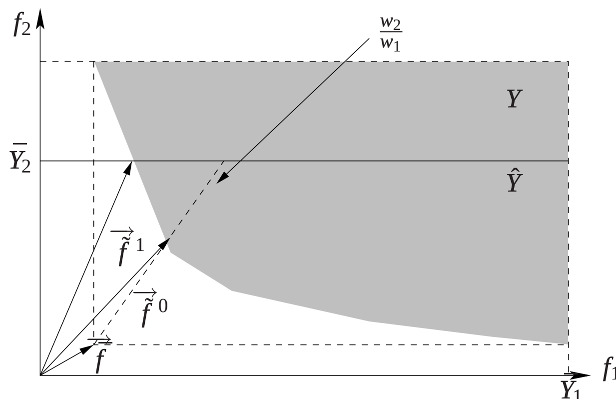


FIG. 3.7 – La méthode S.T.E.P..

A l'issue de cette étape, si l'utilisateur accepte cette solution, le programme s'arrête. Autrement, il faut recommencer.

3.4.3 Discussion

On retrouve la remarque formulée dans la section 3.3.3, à propos de la méthode de Fandel. De plus, cette méthode nécessite de fonctionner avec des fonctions objectif non nulles. Cette dernière remarque limite le champ d'application de cette méthode.

3.5 La méthode de Jahn

3.5.1 Principe

Cette méthode est complètement différente des deux autres. On commence par choisir un point de départ, puis le problème est résolu pas à pas à travers l'espace de recherche en direction de la solution optimale au sens de Pareto [Eschenauer et al. 90].

Cette méthode est fondée sur une méthode de minimisation cyclique (aussi appelée "méthode d'optimisation scalaire dans les directions réalisables" de Zoutendjik).

3.5.2 Présentation de la méthode

On part du problème P.

On transforme ce problème de la manière suivante :

$$\left|\begin{array}{ll} \text{minimiser} & \\ \text{avec} & w_j \cdot \left[\nabla_{\overrightarrow{x}} f_j^u\right]^t \cdot \overrightarrow{\Delta_f} \leq \\ & v_i \cdot \left[\nabla_{\overrightarrow{x}} g_i^u\right]^t \cdot \overrightarrow{\Delta_g} \leq \\ \text{et} & t_l \cdot \left[\nabla_{\overrightarrow{x}} h_l^u\right]^t \cdot \overrightarrow{\Delta_h} = \end{array}\right.$$

On a aussi $\overrightarrow{\Delta_f} \in \mathbb{R}^k$, $\overrightarrow{\Delta_g} \in \mathbb{R}^m$, $\overrightarrow{\Delta_h} \in \mathbb{R}^p$.

$$\text{avec} \quad \begin{array}{l} i = \{1, \cdots, m\} \\ j = \{1, \cdots, k\} \\ l = \{1, \cdots, p\} \\ n = \{1, \cdots, k\} \end{array}$$

On a utilisé les notations suivantes :

:	fonction scalaire de préférence.
$\nabla_{\overrightarrow{x}}$:	opérateur nabla de la variable $\overrightarrow{x}$ ($\nabla_{\overrightarrow{x}} A = \oplus_i \frac{\partial A}{\partial x_i} \cdot \overrightarrow{x}_i$).
$\overrightarrow{\Delta}$:	direction du pas de minimisation.
w_j :	coefficient de pondération de la j^e fonction objectif.
v_i :	coefficient de pondération de la i^e contrainte d'inégalité.
t_l :	coefficient de pondération de la l^e contrainte d'égalité.

La solution de notre nouveau problème commence par le choix d'un vecteur de départ réalisable $\overrightarrow{x}^0$ et par le calcul du vecteur des fonctions objectif correspondant $\overrightarrow{f}\left(\overrightarrow{x}^0\right)$.

Le choix du point de départ influence les alternatives futures possibles, en supposant que l'espace de recherche $\hat{Y}^0$ soit restreint à :

$$\hat{Y}^0 = \left\{\overrightarrow{f}\left(\overrightarrow{x}\right) \mid \overrightarrow{f}\left(\overrightarrow{x}\right) \leq \overrightarrow{f}\left(\overrightarrow{x}^0\right)\right\} \tag{3.12}$$

On a représenté à la figure 3.8 un schéma qui explique ce mécanisme. Sur cette figure, on voit que le choix du vecteur réalisable de départ $\overrightarrow{f}\left(\overrightarrow{x}^0\right)$ permet de restreindre l'espace de recherche. On obtient alors le domaine réalisable $\hat{Y}^0$.

En fonction du vecteur objectif $\overrightarrow{f}\left(\overrightarrow{x}\right)$, l'utilisateur choisit une fonction objectif f_i, qui doit être minimisée prioritairement. A cet effet, un pas de direction $\overrightarrow{\Delta}$ et de longueur appropriée doit être déterminé, et il doit satisfaire la relation suivante :

$$\overrightarrow{x^{k+1}} = \overrightarrow{x^k} + {}^{k} \cdot \overrightarrow{\Delta^k} \tag{3.13}$$

On a utilisé les notations suivantes :

$\overrightarrow{x^{k+1}}$:	variable de travail à l'itération $k+1$.
$\overrightarrow{x^k}$:	variable de travail à l'itération k.
^{k} :	longueur du pas de minimisation.

Il faut que $\overrightarrow{x^{k+1}}$ et $\overrightarrow{x^k}$ satisfassent les contraintes du problème ($\overrightarrow{g}$ et $\overrightarrow{h}$).

De plus, il faut que ces points vérifient :

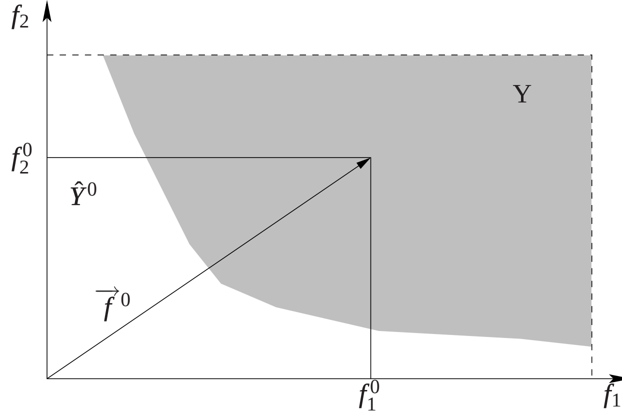


FIG. 3.8 – Initialisation de la méthode de Jahn.

$$
\begin{array}{ll}
f_i^{k+1} < f_i^k & \forall i \in I = \{1, \cdots, k\} \\
f_j^{k+1} \leq f_j^k & \forall j \in J = \{I \mid j \neq i\}
\end{array}
$$

Ces deux relations expriment que la valeur de la fonction objectif calculée à l'itération $k+1$ doit dominer la valeur de la fonction objectif calculée à l'itération k.

On obtient la direction du pas en résolvant notre problème modifié (le problème de départ transformé). En agissant ainsi, les coefficients de pondération w_j, v_i et t_l choisis par l'utilisateur influencent la direction du pas. L'utilisateur peut maintenant choisir la longueur du pas ^{k} dans un intervalle $[0, \ _{max}]$. On trouve $\ _{max}$ en résolvant le problème suivant :

$$
\left|
\begin{array}{lll}
\text{maximiser} & \ _{max} & \\
\text{avec} & f_i\left(\overrightarrow{x^k} + \ _{max} \cdot \overrightarrow{\Delta^k}\right) < f_i\left(\overrightarrow{x^k}\right) & \forall i \in I = \{1, \cdots, k\} \\
 & f_j\left(\overrightarrow{x^k} + \ _{max} \cdot \overrightarrow{\Delta^k}\right) \leq f_j\left(\overrightarrow{x^k}\right) & \forall j \in J = \{I \mid j \neq i\} \\
 & \overrightarrow{g}\left(\overrightarrow{x^k} + \ _{max} \cdot \overrightarrow{\Delta^k}\right) \leq 0 & \\
\text{et} & \overrightarrow{h}\left(\overrightarrow{x^k} + \ _{max} \cdot \overrightarrow{\Delta^k}\right) = 0 &
\end{array}
\right.
$$

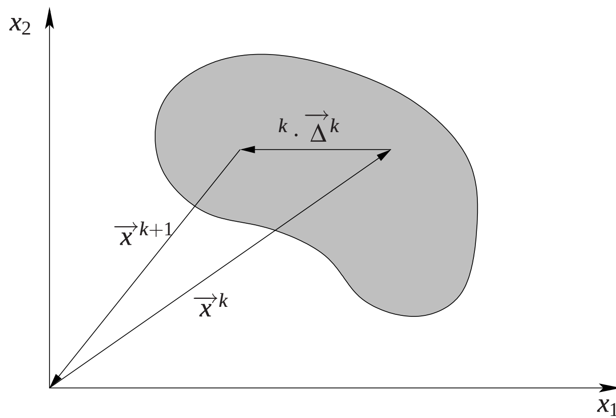
Le nouveau vecteur fonction objectif $\overrightarrow{f^{k+1}}$, pour le pas de direction et de longueur données, est calculé ($\overrightarrow{f^{k+1}} = \overrightarrow{f}\left(\overrightarrow{x^k} + \ ^k \cdot \overrightarrow{\Delta^k}\right)$). Si ce vecteur objectif n'est pas accepté, on détermine à la prochaine itération une nouvelle direction, en résolvant notre problème modifié.

La variation de la direction du pas, qui est faite en fonction des préférences de l'utilisateur à chaque itération, fournit des pas de minimisation à l'intérieur de l'espace de recherche Y, dans la direction de la solution optimale au sens de Pareto $\overrightarrow{f}$.

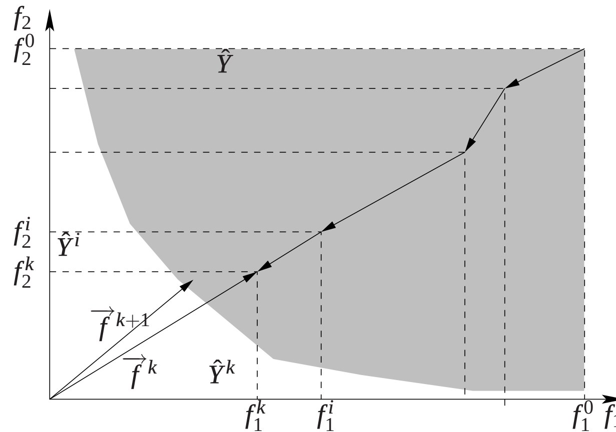
Si l'utilisateur accepte le dernier vecteur objectif $\overrightarrow{f^k}$, l'algorithme s'arrête. Le vecteur $\overrightarrow{f^k}$ n'est pas forcément optimum au sens de Pareto, mais il réside à l'intérieur de l'espace de recherche $\hat{Y}^k$ (l'espace de recherche à l'itération k). Aussi, un point optimum au sens de Pareto est obtenu au voisinage de cette solution en résolvant le problème de substitution scalaire suivant :

$$\left|\begin{array}{ll} \text{minimiser} & \oplus_{i=1}^{k} w_i \cdot \overrightarrow{f}\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \\ \text{et} & \overrightarrow{f}\left(\overrightarrow{x}\right) \leq \overrightarrow{f}\left(\overrightarrow{x^k}\right) \end{array}\right.$$

Un schéma du processus itératif dans X et dans Y est dessiné à la figure 3.9. Sur cette figure, on note $\overrightarrow{\tilde{f}}\left(\overrightarrow{x}\right)$ la solution du problème de substitution scalaire précédent. On peut voir que le vecteur $\overrightarrow{f^k}$ permet de restreindre l'espace de recherche.



(a) Représentation dans X



(b) Représentation dans Y

FIG. 3.9 – La représentation de la méthode de Jahn.

Pour illustrer la dernière étape de la méthode, la figure 3.10 représente la résolution du problème de substitution scalaire (méthode de pondération des fonctions objectif).

3.5.3 Discussion

On retrouve l'inconvénient signalé dans la section 3.3.3, à propos de la méthode de Fandel. De plus, cette méthode a besoin de connaître les gradients de certaines expressions

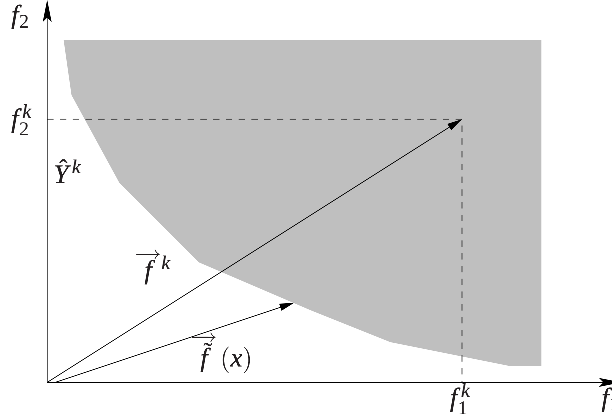


FIG. 3.10 – Dernière étape de la méthode de Jahn.

liées au problème d'optimisation : gradient du vecteur fonction objectif, gradient du vecteur contrainte. Par conséquent, si le problème n'admet pas de dérivées en certains points (comme les problèmes combinatoires), cette méthode ne peut pas être appliquée.

3.6 La méthode de Geoffrion

3.6.1 Principe

Cette approche est similaire à la méthode de Jahn. Elle repose sur un algorithme : l'algorithme de Franck-Wolfe [Eschenauer et al. 90].

3.6.2 Présentation de la méthode

On part du problème P.

On transforme le problème de la manière suivante :

$$\left|\begin{array}{lll} \text{minimiser} & \left(\nabla_{\overrightarrow{x}}\, p\left[\overrightarrow{f}\left(\overrightarrow{x}\right)\right]\right)^t \cdot \overrightarrow{\Delta} & \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 & \text{problème } P' \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 & \end{array}\right.$$

avec $\overrightarrow{\Delta} \in \mathbb{R}^k$

La fonction de préférence $p\left[\overrightarrow{f}\left(\overrightarrow{x}\right)\right]$ n'est pas forcément connue de l'utilisateur. Pendant le déroulement de l'algorithme, l'utilisateur devra fournir des informations sur les compromis qu'il désire réaliser. Il doit aussi fournir la longueur de pas.

Deux méthodes sont possibles pour déterminer le point de départ $\overrightarrow{x^0}$. Soit l'utilisateur sélectionne un vecteur en utilisant la méthode de Jahn, soit le vecteur est calculé en résolvant le problème de substitution suivant :

$$\left|\begin{array}{ll} \text{minimiser} & \bigoplus_{i=1}^{k} w_i \cdot f_i\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

On reconnaît là la méthode de pondération des fonctions objectif décrite dans la section 2.1.2.

En résolvant ce problème, on obtient le point $\overrightarrow{x^0} = \overrightarrow{x^*}$.

A ce moment, on peut résoudre le problème P'. Cette résolution nous donne la direction de recherche Δ. Des informations à propos de la fonction de préférence sont maintenant requises. Il va falloir transformer le problème P' en utilisant la relation suivante :

$$\nabla_{\overrightarrow{x}}\, p\left[\overrightarrow{f}\left(\overrightarrow{x}\right)\right] = \bigoplus_{i=1}^{k} \left[\frac{\partial p}{\partial f_i}\right]_{\overrightarrow{x}} \cdot \nabla_{\overrightarrow{x}}\, f_i\left(\overrightarrow{x}\right) \tag{3.14}$$

On obtient alors le problème suivant :

$$\left|\begin{array}{ll} \text{minimiser} & \oplus_{i=1}^{k} w_i \cdot \left(\nabla_{\overrightarrow{x}}\, f_i\left(\overrightarrow{x}^k\right)\right)^t \cdot \overrightarrow{\Delta} \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

On a :

$$w_i = \frac{\left[\frac{\partial p}{\partial f_i}\right]_{\overrightarrow{x}}}{\left[\frac{\partial p}{\partial f_1}\right]_{\overrightarrow{x}}} \tag{3.15}$$

avec $i = \{1, \cdots, k\}$

et la direction du pas est donnée par $\overrightarrow{\Delta^k} = \overrightarrow{x^{k+1}} - \overrightarrow{x^k}$.

Les coefficients de pondération reflètent le compromis qui est fait par l'utilisateur entre la fonction objectif f_i et la fonction objectif de référence f_1.

Il y a deux méthodes pour déterminer les rapports de compromis w_i sans faire appel à l'utilisateur. La première méthode consiste à autoriser une variation infinitésimale Δf_i, qui compense la variation infinitésimale Δf_1 de la fonction objectif f_1, en gardant les autres fonctions objectif inchangées. Alors :

$$w_i = -\frac{\Delta f_1}{\Delta f_i} \tag{3.16}$$

L'autre méthode consiste à "approximer" les coefficients de pondération en faisant l'hypothèse que le gradient de $p\left[\overrightarrow{f}\left(\overrightarrow{x}\right)\right]$ reste constant dans la direction du pas de recherche. Par conséquent, si tous les objectifs, exceptés le premier et le i^e, sont maintenus constants, la fonction de préférence inconnue $p\left[\overrightarrow{f}\left(\overrightarrow{x}\right)\right]$ s'accroît plus rapidement quand la fonction objectif i varie de b_i unités en proportion de la variation de b_1 unités de la fonction objectif de référence 1 :

$$w_i = \frac{b_i}{b_1} \tag{3.17}$$

Après résolution du problème de substitution ci-dessus, la longueur du pas   doit être déterminée par l'utilisateur lui-même, qui doit considérer le but :

$$\left|\begin{array}{ll} \text{minimiser} & p\left[f\left(\overrightarrow{x^k} + {}^{k} \cdot \overrightarrow{\Delta^k}\right)\right] \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x^k} + {}^{k} \cdot \overrightarrow{\Delta^k}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x^k} + {}^{k} \cdot \overrightarrow{\Delta^k}\right) = 0 \end{array}\right.$$

en utilisant l'hypothèse suivante :

$$0 \leq {}^{k} \leq 1$$

La figure 3.11 représente le fonctionnement de cet algorithme. Sur cette figure, on a noté $\overrightarrow{\tilde{f}^{\rangle}}$ la solution trouvée à l'itération i. Le premier point sur cette figure ($\overrightarrow{F^0}$) a été obtenu par résolution du problème de substitution par pondération des fonctions objectif. On a $\overrightarrow{F^0} = \overrightarrow{f}\left(\overrightarrow{x^0}\right)$.

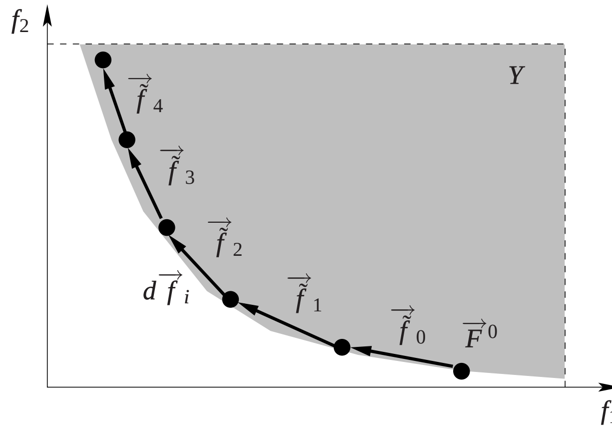


FIG. 3.11 – La méthode de Geoffrion.

Pendant la minimisation ci-dessus, l'utilisateur peut être aidé par une représentation graphique de toutes les fonctions objectif en fonction de , ou par une liste de dépendance sous forme de tableau.

Une fois que la longueur de pas est déterminée, le nouveau vecteur objectif peut être calculé :

$$\overrightarrow{f^{k+1}} = \overrightarrow{f}\left(\overrightarrow{x} + {}^{k} \cdot \overrightarrow{\Delta^k}\right) \tag{3.18}$$

L'utilisateur évalue alors le résultat, et décide si le processus peut continuer, ou si la solution peut être acceptée comme solution de compromis. S'il décide de continuer, il peut alors parcourir la surface de compromis (voir figure 3.11).

3.6.3 Discussion

On retrouve l'inconvénient habituel des méthodes interactives (voir section 3.3.3). Encore une fois, soulignons la restriction de cette méthode à des problèmes pour lesquels on peut calculer la dérivée.

3.7 La méthode du simplex

3.7.1 Principe

Cette méthode n'a rien à voir avec la méthode du simplexe utilisée en programmation linéaire. Il s'agit là d'une méthode de recherche séquentielle [Nelder et al. 65] de l'optimum d'un problème d'optimisation. Le logiciel [Multisimplex] réalise cette optimisation pour un problème d'optimisation multiobjectif de manière interactive.

Cette méthode utilise $k+1$ essais (où k représente la dimension de la variable de décision $\overrightarrow{x}$) pour définir une direction d'amélioration des fonctions objectif.

L'amélioration des fonctions objectif est obtenue en utilisant une méthode d'agrégation floue (voir section 4.1).

3.7.2 Présentation de la méthode

On commence par choisir au hasard $k+1$ valeurs pour la variable de décision $\overrightarrow{x}$. L'algorithme évalue alors les $k+1$ points et supprime le point le moins "efficace". Il crée alors un nouveau point à partir du point supprimé (voir figure 3.12) et recommence l'évaluation.

L'algorithme comporte quelques règles, qui permettent de choisir des points qui évitent de tourner autour d'une mauvaise solution. Les deux principales règles sont les suivantes :

Règle 1 : **rejeter les pires solutions.**
Une nouvelle position de la variable de décision $\overrightarrow{x}$ est calculée par réflexion de la position rejetée (voir la figure 3.12). Après cette transformation, on recherche le nouveau pire point. La méthode est alors répétée en éliminant ce point, etc. A chaque étape, on se rapproche de la zone où se trouve l'optimum recherché.

Règle 2 : **ne jamais revenir sur un point qui vient juste d'être rejeté.**
Sans cette règle, l'algorithme pourrait osciller entre deux "mauvais" points.

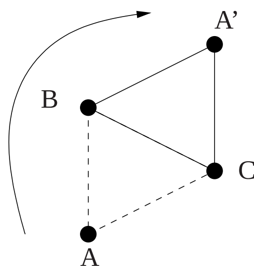


FIG. 3.12 – Choix d'un nouveau point par symétrie.

Sur la figure 3.13, on peut voir le cheminement de la méthode dans l'espace des variables de décision, dans le cas d'un exemple à deux variables de décision. Sur cette figure sont représentées des courbes de niveau représentant la "direction" d'amélioration de la fonction objectif à optimiser. Dans cet exemple, l'optimisation se déroule de manière à maximiser le pourcentage correspondant à la courbe de niveau.

Présentons maintenant l'algorithme complet de la méthode du simplex. Nous utiliserons les notations suivantes :

W : point le moins favorable ou point venant d'être rejeté.

B : point le plus favorable.

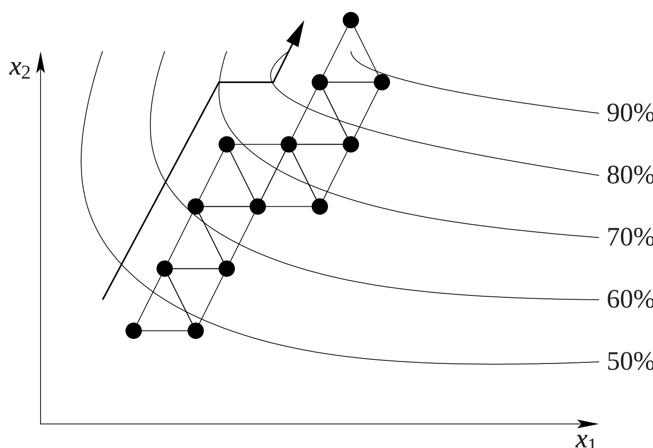


FIG. 3.13 – Cheminement de la méthode du simplex.

N : second point le plus favorable.

L'algorithme du simplex est présenté à la figure 3.14.

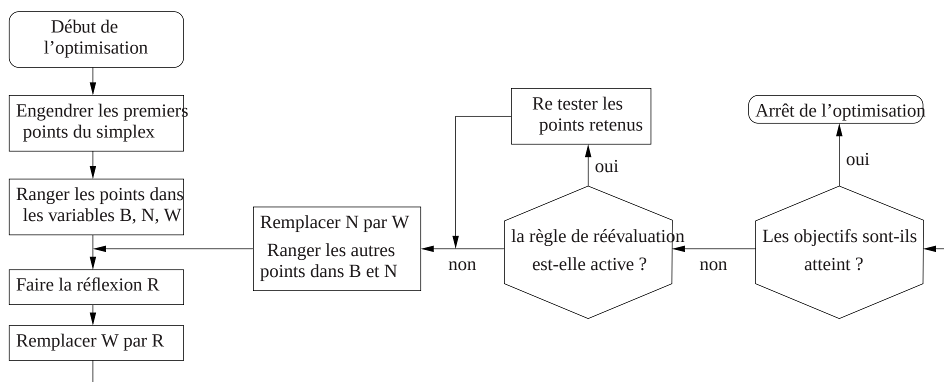


FIG. 3.14 – L'algorithme de la méthode du simplex.

Il existe aussi une version modifiée de la méthode du simplex. Cette méthode, dite du simplex modifié, comporte deux règles supplémentaires :

Règle 3 : dilater le déplacement dans le sens de l'amélioration des fonctions objectif.

Règle 4 : contracter le déplacement, si celui-ci a été fait dans une direction qui n'est pas favorable à l'amélioration des fonctions objectif.

Sur un problème à deux variables de décision, on obtient le schéma situé à la figure 3.15.

En reprenant les mêmes notations que pour l'algorithme précédent, nous obtenons la représentation de la figure 3.16.

Pour les différentes transformations, on a les expressions suivantes :

Réflexion : $R = \overline{C} + \quad \cdot \left(\overline{C} - W\right)$

Dilatation : $E = \overline{C} + \quad \cdot \left(\overline{C} - W\right)$

Contraction positive : $C_{+} = \overline{C} + \quad ^{+} \cdot \left(\overline{C} - W\right)$

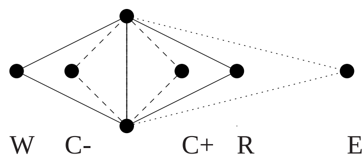


FIG. 3.15 – Le processus de dilatation et de contraction de la méthode du simplex modifié.

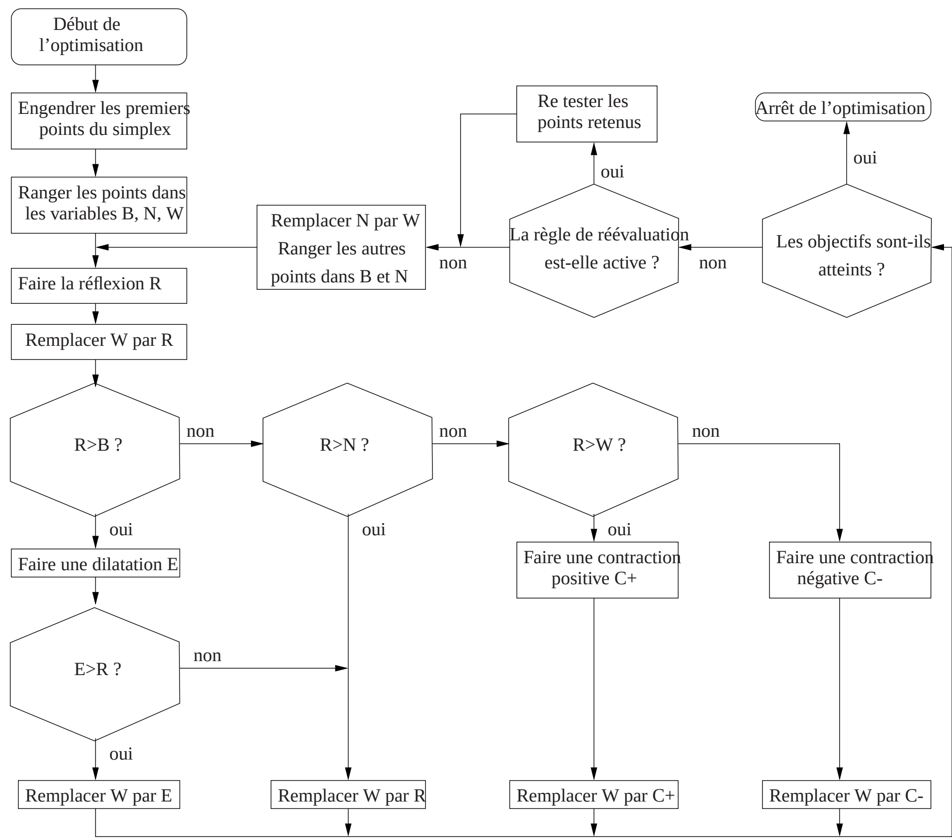


FIG. 3.16 – L'algorithme de la méthode du simplex modifié.

Contraction négative : $C_{-} = \bar{C} - \;{}^{-} \cdot \left(\bar{C} - W\right)$

où

W	est le point rejeté,
$\bar{C}$	est le barycentre des points restants,
	est le coefficient de réflexion ($= 1$ par défaut),
	est le coefficient de dilatation ($= 2$ par défaut),
$^{+}$	est le coefficient de contraction positive ($^{+} = 0.5$ par défaut),
$^{-}$	est le coefficient de contraction négative ($^{-} = 0.5$ par défaut).

3.7.3 Discussion

Cette méthode est très simple à mettre en œuvre. Elle a, d'ailleurs, fait l'objet d'une réalisation industrielle sous la forme du logiciel Multisimplex (voir l'URL cité à la référence [Multisimplex]).

3.8 Bibliographie commentée

[Eschenauer et al. 90] C'est un des rares ouvrages discutant de méthodes interactives. L'exposé de ces méthodes est très mathématique et ces méthodes ne sont pas illustrées à travers des exemples simples. En revanche, on trouvera dans les derniers chapitres de nombreux exemples d'applications de l'optimisation multiobjectif sur des problèmes industriels ; en particulier un problème de recherche de la structure optimale d'une antenne parabolique visant à minimiser les vibrations de la structure.

[Miettinen 99] On trouvera aussi la description d'un certain nombre de méthodes d'optimisation multiobjectif interactives dans cet ouvrage.

Chapitre 4

Les méthodes floues

4.1 Introduction à la logique floue

Dans la vie courante, tout ne peut pas être décrit de manière binaire. Par exemple, la transition entre le jour et la nuit se fait progressivement, l'action sur l'embrayage d'un véhicule est, elle aussi, progressive.

Longtemps, le seul outil de description en logique était binaire. Tout en logique a été décrit en termes de VRAI ou FAUX. Le problème de cette description simpliste en logique est qu'elle ne permet pas de traiter l'incertitude et l'imprécision des connaissances humaines.

L'automaticien L. A. Zadeh a élaboré une nouvelle logique basée sur les ensembles flous. Elle permet de traiter l'imprécision et l'incertitude dans la connaissance humaine ainsi que les transitions progressives entre états.

La différence principale entre une logique classique et cette logique floue est l'existence d'une transition progressive entre le VRAI et le FAUX [Zadeh 65, Remez 86, Vern 92].

4.1.1 Parallèle avec la logique classique

Pour mieux comprendre le concept de la logique floue, nous allons transposer les concepts de la logique classique en logique floue.

Tout d'abord, la logique classique possède trois fonctions de base. Ces fonctions travaillent sur des paramètres binaires (0 ou 1, FAUX ou VRAI). Voici ces fonctions de base :

– La fonction ET : $C = A\, et\, B$, ou, en termes plus mathématiques, $C = A \cdot B$. La table de vérité de cette fonction est représentée dans le tableau 4.1.

A \ B	0	1
0	0	0
1	0	1

TAB. 4.1 – La fonction ET.

A \ B	0	1
0	0	1
1	1	1

TAB. 4.2 – La fonction OU.

A	C
0	1
1	0

TAB. 4.3 – La fonction NON.

– La fonction OU : $C = A$ ou B ou, en termes plus mathématiques, $C = A + B$. La table de vérité de cette fonction est représentée dans le tableau 4.2.
– La fonction NON : $C =$ non A ou, en termes plus mathématiques, $C = \overline{A}$. La table de vérité de cette fonction est représentée dans le tableau 4.3.

Toutes ces fonctions de la logique classique ont leur équivalent en logique floue. La grande différence vient du fait que l'on travaille sur une variable logique qui varie de manière continue de 0 à 1.

Les équivalents en logique floue des fonctions de la logique classique sont représentés dans le tableau 4.4.

Logique classique	Logique floue
$C = A$ et B	$C = \min(A, B)$
$C = A$ ou B	$C = \max(A, B)$
$C =$ non A	$C = 1 - A$

TAB. 4.4 – Equivalence entre logique classique et logique floue.

4.1.2 La fonction d'appartenance

En général, en logique floue, on rattache toujours la variable (A, B ou C) à une donnée extérieure (une information ou une mesure, par exemple). Ce rattachement s'effectue par l'intermédiaire de ce que l'on appelle une fonction d'appartenance. Cette fonction permet de faire correspondre une information (par exemple la taille d'un individu variant de 0 à 2 m) à une variable logique continue variant de 0 à 1. Voici un exemple de fonction d'appartenance pour une taille. Cette fonction définit la variable logique floue "taille moyenne". On considère que l'on a une taille moyenne lorsque celle-ci varie entre 1.70 m et 1.80 m. Les figures 4.1 et 4.2 illustrent les différences existant sur cet exemple entre logique floue et logique classique.

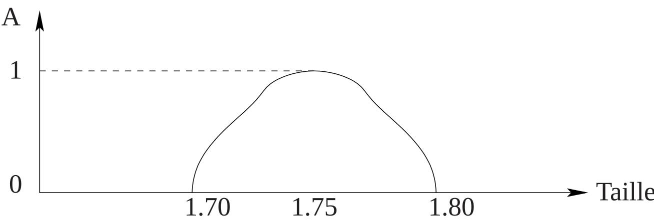


FIG. 4.1 – Fonction d'appartenance "Taille moyenne" en logique floue.

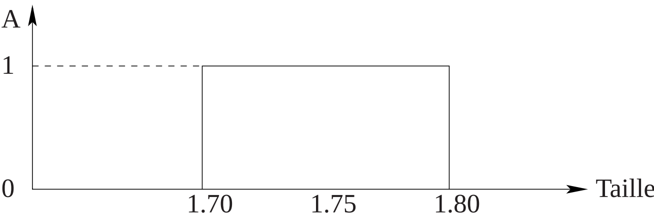


FIG. 4.2 – Fonction "Taille moyenne" en logique classique.

Les différentes fonctions de logique floue que nous venons de présenter possèdent les mêmes propriétés que les fonctions de logique classique (distributivité, commutativité et règles de De Morgan). Une transposition de ces règles en logique floue est représentée dans le tableau 4.5.

Logique classique	Logique floue
distributivité	
$(A+B)\cdot C = A\cdot C + B\cdot C$	$\min(\max(A,B),C) = \max(\min(A,C),\min(B,C))$
commutativité	
$(A+B)+C = A+(B+C)$	$\max(\max(A,B),C) = \max(A,\max(B,C))$
règle de De Morgan	
$\overline{A\cdot B} = \overline{A}+\overline{B}$	$1-\min(A,B) = \min(1-A,1-B)$

TAB. 4.5 – Les propriétés des deux logiques.

Non allons, pour finir, combiner ces différentes fonctions floues pour les variables floues “Taille moyenne” (entre 1.70 et 1.80) et “Taille grande” (entre 1.75 et 1.85).

– La fonction ET (fonction ***min*** en logique floue, voir figure 4.3).

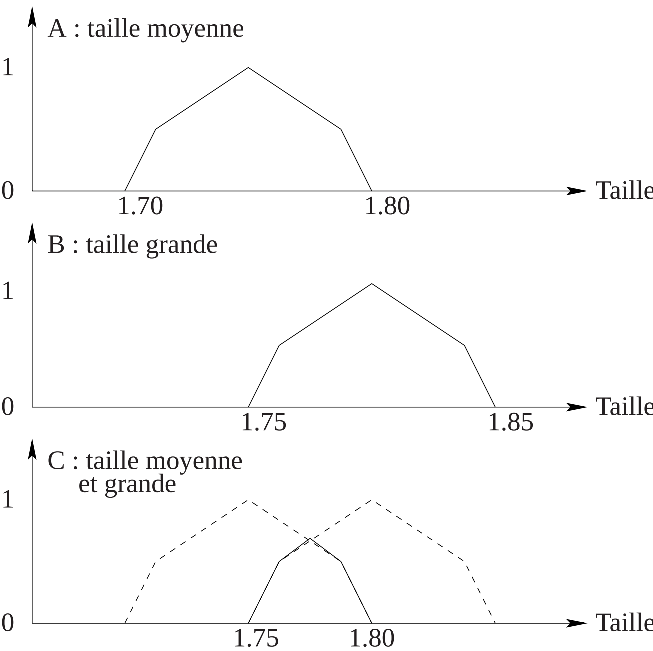


FIG. 4.3 – La fonction ET de logique floue.

– La fonction OU (fonction ***max*** en logique floue, voir figure 4.4).
– La fonction NON (fonction complément à 1 en logique floue, voir figure 4.5).

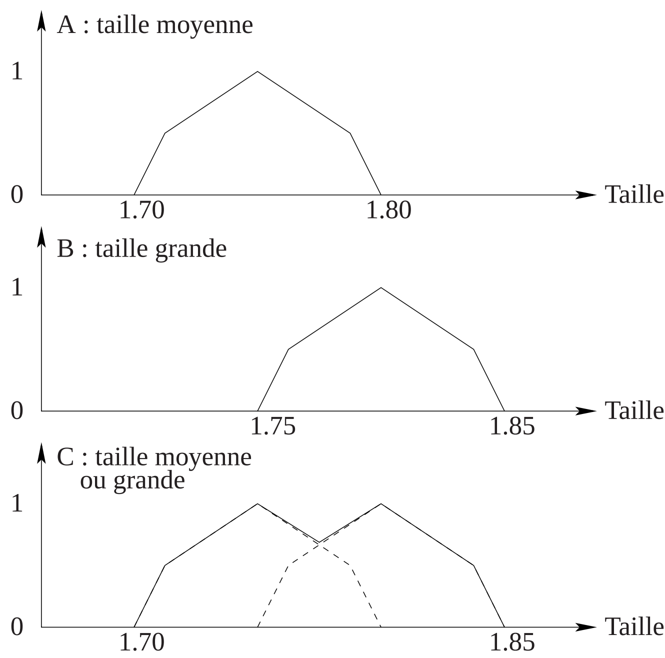


FIG. 4.4 – La fonction OU de logique floue.

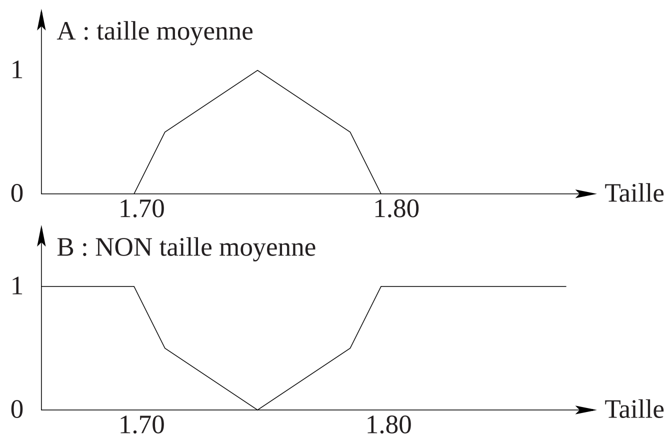


FIG. 4.5 – La fonction NON de logique floue.

4.2 La méthode de Sakawa

4.2.1 Principe

Cette méthode fait intervenir la logique floue à tous les niveaux (sur les paramètres du problème ainsi que sur les paramètres des contraintes). L'ensemble de solutions que l'on va trouver, lui aussi, fera intervenir la logique floue. Ces solutions auront un niveau d'appartenance. Ce seront des solutions qui auront une ***corrélation*** variable avec l'objectif initial

[Sakawa et al. 88] et [Sakawa 02] (le niveau de corrélation avec l'objectif initial est fixé par l'utilisateur).

4.2.2 Présentation de la méthode

Cette méthode d'optimisation avec traitement par la logique floue est présentée en utilisant un problème d'optimisation linéaire (encore appelé problème de programmation linéaire). On part du problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & \overrightarrow{c}_1 \cdot \overrightarrow{x}^t, \cdots, \overrightarrow{c}_k \cdot \overrightarrow{x}^t \\ \text{avec} & \oplus_i a_{ij} \cdot x_i \leq b_j,\ j = \{1, \cdots, m\} \end{array} \right.$$

On a $\overrightarrow{x} \in \mathbb{R}^n$, $\overrightarrow{c} \in \mathbb{R}^n$, $\overrightarrow{a} \in \mathbb{R}^n$ et $\overrightarrow{b} \in \mathbb{R}^m$.

On désigne par :

n : le nombre de variables du problème (dimension de $\overrightarrow{x}$).

k : le nombre de fonctions objectif.

m : le nombre de contraintes d'inégalité.

On appelle X l'ensemble suivant :

$$X = \left\{ \overrightarrow{x} \in \mathbb{R}^n \mid \bigoplus_{i=1}^{n} a_{ij} \cdot x_i \leq b_j,\ j = \{1, \cdots, m\} \right\} \tag{4.1}$$

C'est l'ensemble des valeurs de $\overrightarrow{x}$ telles que les contraintes soient respectées.

Pour ce problème, on estime que l'on a une incertitude sur les fonctions objectif. On va donc considérer que les paramètres sont flous. On récrit le problème de la manière suivante :

$$\left| \begin{array}{ll} \text{minimiser} & \overrightarrow{\tilde{c}_1} \cdot \overrightarrow{x}^t, \cdots, \overrightarrow{\tilde{c}_k} \cdot \overrightarrow{x}^t \\ \text{avec} & \overrightarrow{x} \in X\left(\tilde{a}, \tilde{b}\right) \end{array} \right.$$

Ici, on note :

$$X\left(\tilde{a}, \tilde{b}\right) = \left\{ \overrightarrow{x} \in \mathbb{R}^n \mid \bigoplus_{i=1}^{n} \tilde{a}_{ij} \cdot x_i \leq \tilde{b}_j,\ j = \{1, \cdots, m\} \right\} \tag{4.2}$$

Pour chaque paramètre, on va définir une fonction d'appartenance :

$$\mu_{\tilde{c}_{j1}}\left(c_{j1}\right), \cdots, \mu_{\tilde{c}_{jn}}\left(c_{jn}\right) ; \mu_{\tilde{b}_1}\left(b_1\right), \cdots, \mu_{\tilde{b}_n}\left(b_n\right) ; \mu_{\tilde{a}_{i1}}\left(a_{i1}\right), \cdots, \mu_{\tilde{a}_{in}}\left(a_{in}\right)$$

Avant de continuer dans la présentation de la méthode, nous allons introduire deux définitions.

Définition 1 : ensemble de niveau
L'ensemble de niveau des nombres flous $\tilde{a}_{jr}$, $\tilde{b}_j$, $\tilde{c}_{jr}$ est défini comme l'ensemble $L\ \left(\tilde{a}, \tilde{b}, \tilde{c}\right)$ pour lequel le degré d'appartenance des fonctions $\tilde{a}$, $\tilde{b}$ et $\tilde{c}$ est supérieur ou égal au niveau :

$$L\ \left(\tilde{a}, \tilde{b}, \tilde{c}\right) = \left\{ (a, b, c) \mid \mu_{\tilde{c}_{jr}}\left(c_{jr}\right) \geq\ ,\ \mu_{\tilde{b}_j}\left(b_j\right) \geq\ \text{ et } \mu_{\tilde{a}_{ir}}\left(a_{ir}\right) \geq\ \right\} \tag{4.3}$$

avec $i = \{1, \cdots, k\}$, $j = \{1, \cdots, m\}$ et $r = \{1, \cdots, n\}$.
Cet ensemble possède la propriété suivante :

$$_1 \leq \;\; _2 \text{ si } L_{\;_1}\left(\tilde{a}, \tilde{b}, \tilde{c}\right) \supset L_{\;_2}\left(\tilde{a}, \tilde{b}, \tilde{c}\right) \tag{4.4}$$

> **Définition 2 : solution -optimale au sens de Pareto**
> $\overrightarrow{x}^* \in X\left(\tilde{a}, \tilde{b}\right)$ est appelé une solution -optimale au sens de Pareto du problème de programmation linéaire multiobjectif si et seulement s'il n'existe pas de $\overrightarrow{x} \in X\left(\tilde{a}, \tilde{b}\right)$ et de $(a, b, c) \in L\;\;\left(\tilde{a}, \tilde{b}, \tilde{c}\right)$ tels que $\overrightarrow{c}_i \cdot \overrightarrow{x} \leq \overrightarrow{c}_i^* \cdot \overrightarrow{x}$, $i = \{1, \cdots, k\}$ avec une inégalité stricte pour au moins un i. Les paramètres (a^*, b^*, c^*) sont appelés paramètres optimaux de niveau .

Nous allons maintenant déterminer l'allure des fonctions d'appartenance $\mu_i(\overrightarrow{c}_i \cdot \overrightarrow{x})$, $i = \{1, \cdots, k\}$. On commence par calculer les valeurs extrêmales des fonctions objectif sous les contraintes $\;= 0$ et $\;= 1$. On note ces valeurs $(\overrightarrow{c}_i \cdot \overrightarrow{x})^0$ et $(\overrightarrow{c}_i \cdot \overrightarrow{x})^1$.

- $(\overrightarrow{c}_i \cdot \overrightarrow{x})^0$ est une valeur inacceptable pour $(\overrightarrow{c}_i \cdot \overrightarrow{x})$;
- $(\overrightarrow{c}_i \cdot \overrightarrow{x})^1$ est une valeur très acceptable pour $(\overrightarrow{c}_i \cdot \overrightarrow{x})$.

Donc, on aura :

$$\mu_i(\overrightarrow{c}_i \cdot \overrightarrow{x}) = \begin{cases} 1, \text{ ou tend vers } 1 & \text{si } (\overrightarrow{c}_i \cdot \overrightarrow{x})^1 \geq (\overrightarrow{c}_i \cdot \overrightarrow{x}) \\ d_i(\overrightarrow{c}_i \cdot \overrightarrow{x}) & \text{si } (\overrightarrow{c}_i \cdot \overrightarrow{x})^1 \leq (\overrightarrow{c}_i \cdot \overrightarrow{x}) \leq (\overrightarrow{c}_i \cdot \overrightarrow{x})^0 \\ 0, \text{ ou tend vers } 0 & \text{si } (\overrightarrow{c}_i \cdot \overrightarrow{x}) \geq (\overrightarrow{c}_i \cdot \overrightarrow{x})^0 \end{cases} \tag{4.5}$$

où $d_i(\overrightarrow{c}_i \cdot \overrightarrow{x})$ est une fonction monotone strictement décroissante, qui peut être linéaire ou non linéaire.

Maintenant que les fonctions d'appartenance des fonctions objectif sont définies, nous allons pouvoir récrire notre problème, en utilisant le formalisme flou :

$$\left| \begin{array}{ll} \text{maximiser} & \mu_D(\mu_1(\overrightarrow{c}_1 \cdot \overrightarrow{x}), \cdots, \mu_k(\overrightarrow{c}_i \cdot \overrightarrow{x})) \\ \text{avec} & \overrightarrow{x} \in X\left(\tilde{a}, \tilde{b}\right) \\ \text{et} & (a, b, c) \in L\;\;\left(\tilde{a}, \tilde{b}, \tilde{c}\right) \end{array} \right.$$

où μ_D est une fonction d'agrégation. μ_D peut être interprétée comme le degré de satisfaction vis-à-vis du but fixé par l'utilisateur. Un exemple de fonction d'agrégation utilisée est :

$$\min(\mu_1(\overrightarrow{c}_i \cdot \overrightarrow{x}), \cdots, \mu_k(\overrightarrow{c}_i \cdot \overrightarrow{x})) \tag{4.6}$$

On obtient finalement le problème suivant (si l'on utilise la fonction d'agrégation ci-dessus) :

$$\left| \begin{array}{ll} \text{maximiser} & \min(\mu_1(\overrightarrow{c}_i \cdot \overrightarrow{x}), \cdots, \mu_k(\overrightarrow{c}_i \cdot \overrightarrow{x})) \\ \text{avec} & \overrightarrow{x} \in X\left(\widetilde{a}, \widetilde{b}\right) \\ \text{et} & (a, b, c) \in L\;\;\left(\widetilde{a}, \widetilde{b}, \widetilde{c}\right) \end{array} \right.$$

On va donc chercher à maximiser le degré d'appartenance minimal. [Munro et al. 86] décrit une méthode similaire à celle-ci, mais appliquée à un problème d'optimisation multiobjectif général.

Pour résumer, ou peut décomposer cette méthode en six étapes.

On part du problème suivant :

$$\left| \begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}), \cdots, f_k(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \end{array} \right.$$

Etape 1 : sous la contrainte $\overrightarrow{g}(\overrightarrow{x}) \leq 0$, on minimise, puis on maximise, chaque fonction objectif séparément, de manière à obtenir la plage de variation de la fonction d'appartenance des fonctions objectif.

Etape 2 : On définit les fonctions d'appartenance de la manière suivante :

$$\mu_i(\overrightarrow{x}) = \frac{f_{i1} - f_i(\overrightarrow{x})}{f_{i1} - f_{i0}} \tag{4.7}$$

où
- f_{i0} est la valeur de la fonction d'appartenance la moins intéressante,
- f_{i1} est la valeur de la fonction d'appartenance la plus intéressante.

Etape 3 : On définit les fonctions de prise de décision DM_i (*Decision Making*) comme suit :

$$DM_i(\overrightarrow{x}) = \begin{cases} 0 & \text{si } \mu_i(\overrightarrow{x}) \leq 0 \\ \mu_i(\overrightarrow{x}) & \text{si } 0 < \mu_i(\overrightarrow{x}) \leq 1 \\ 1 & \text{si } 1 \leq \mu_i(\overrightarrow{x}) \end{cases} \tag{4.8}$$

où
- $DM_i(\overrightarrow{x}) = 1$ quand l'objectif est atteint,
- $DM_i(\overrightarrow{x}) = 0$ quand l'objectif n'est pas atteint.

Etape 4 : On maximise les fonctions DM_i.

Etape 5 : S'il n'y a pas de solutions, ou si la solution ne satisfait pas l'utilisateur, alors il faut modifier les fonctions d'appartenance, et retourner à l'étape 3.

Etape 6 : Arrêt et affichage des résultats.

Ce résumé est tiré de [Niimura et al.].

4.2.3 Discussion

Cette méthode est très intéressante, car elle permet de prendre en compte l'exigence de l'utilisateur vis-à-vis de la réponse fournie par l'algorithme de recherche. De cette manière, l'utilisateur peut obtenir un plus grand nombre de réponses s'il est moins exigeant.

Cette exigence est symbolisée par le niveau d'appartenance . Si l'utilisateur est très exigeant, il fixera un niveau minimum proche de 1.

Cependant, la méthode de Sakawa semble "lourde" à mettre en œuvre. Elle fait appel à des ensembles flous et à des règles mathématiques difficiles à appliquer. On préférera l'approche utilisée dans la méthode de Reardon que nous allons décrire maintenant.

4.3 La méthode de Reardon

4.3.1 Introduction

On présente une méthode simplifiée de traitement multiobjectif utilisant la logique floue. Des exemples utilisant cette méthode peuvent être trouvés dans [Reardon 97a] et [Reardon 97b].

4.3.2 Présentation de la méthode

On part du problème P.

Pour chaque fonction objectif, on définit une fonction d'appartenance, qui aura la forme représentée à la figure 4.6.

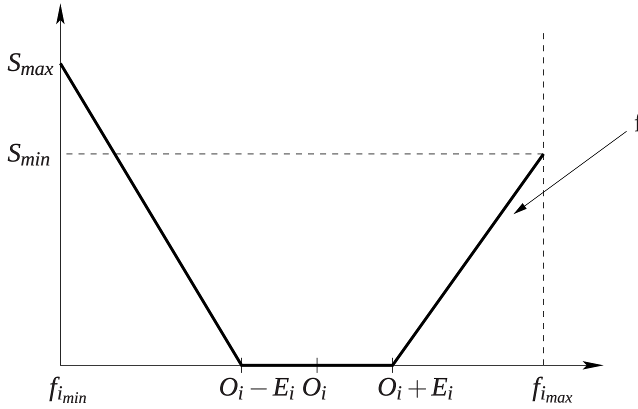


FIG. 4.6 – La fonction d'appartenance de Reardon.

Cette fonction d'appartenance permet d' "informer" l'algorithme que la fonction objectif a sa valeur située dans l'intervalle $[O_i - E_i, O_i + E_i]$ (zone où la fonction d'appartenance vaut 0). Si la valeur de la fonction objectif est située en dehors de cette zone, on pénalise de plus en plus la fonction objectif. Dans ce cas, la pénalité varie entre 0 et S_{min} ou S_{max}.

La signification des différents paramètres de la fonction d'appartenance est la suivante :

S_{min} et S_{max} : facteurs d'échelle flous.

$f_{i_{min}}$: valeur minimale de la fonction objectif numéro *i*.

$f_{i_{max}}$: valeur maximale de la fonction objectif numéro *i*.

O_i : valeur expérimentale de la fonction objectif numéro *i*. On voudrait que $f_i = O_i$.

E_i : marge d'erreur acceptable (l'objectif O_i est défini à E_i près).

D'après le schéma, l'équation de la fonction d'appartenance est la suivante :

– si $f_i \leq (O_i - E_i)$ alors :

$$f^{'}(f_i) = \left[\frac{S_{max}}{f_{i_{min}} - (O_i - E_i)}\right] \cdot (f_i - (O_i - E_i))$$

– si $(O_i - E_i) \leq f_i \leq (O_i + E_i)$ alors :

$$f^{'}(f_i) = 0$$

– si $f_i \geq (O_i + E_i)$ alors :

$$f^{'}(f_i) = \left[\frac{S_{min}}{(O_i + E_i) - f_{i_{max}}}\right] \cdot (f_i - (O_i + E_i))$$

Ici, par rapport à la méthode “normale” d’optimisation floue, le niveau est inversé. Au lieu d’avoir 1 pour la fonction d’appartenance quand f_i appartient à la plage de valeurs désirées (entre $O_i - E_i$ et $O_i + E_i$), la fonction d’appartenance vaut 0. Cette inversion du niveau d’appartenance permet d’obtenir une fonction objectif globale plus simple, qui réunit ou “agrège” les fonctions objectif :

$$F = \frac{1}{N} \cdot \bigoplus_{i=1}^{N} f_i^{'}(f_i) \tag{4.9}$$

Comme on peut le voir sur cette formule, F est une simple moyenne des fonctions d’appartenance des fonctions objectif.

Ces différentes fonctions d’appartenance ont été normalisées par l’intermédiaire de S_{min} et S_{max}. De cette manière, on a pu gommer l’influence de la différence entre les unités de mesure des différentes fonctions objectif, et aussi minimiser les effets dus aux différences de plages de variation entre les fonctions objectif. On a, en quelque sorte, hiérarchisé les différentes fonctions objectif.

Grâce à cet artifice, résoudre notre problème revient à trouver un point qui minimise F. Cela revient donc à trouver un point tel que les $f_i^{'}(f_i)$ soient minimaux, donc tel que le niveau d’appartenance soit le plus élevé possible.

4.3.3 Discussion

Cette méthode est simple à mettre en œuvre. Par exemple, elle donne de bons résultats sur deux problèmes tests (le problème Schaffer’s F2 [Reardon 97a] et le problème simplifié de Born-Mayer [Reardon 97b]).

Chapitre 5

Les méthodes exploitant une métaheuristique

5.1 Qu'est-ce qu'une métaheuristique ?

Les métaheuristiques, déjà évoquées dans l'avant-propos, sont des méthodes générales de recherche dédiées aux problèmes d' “optimisation difficile” [Sait et al. 99]. Ces méthodes sont, en général, présentées sous la forme de concept. Comme nous le verrons plus tard, elles reprennent des idées que l'on retrouve parfois dans la vie courante. Les principales métaheuristiques sont le recuit simulé, la recherche tabou et les algorithmes génétiques.

5.2 Décomposition des tâches en optimisation

On remarque que, la plupart du temps, les méthodes d'optimisation multiobjectif et les métaheuristiques associées peuvent être complètement découplées. Seuls les algorithmes génétiques engendrent des méthodes d'optimisation multiobjectif qui ne peuvent pas être découplées.

On pourra donc diviser l'ensemble “méthode d'optimisation multiobjectif + métaheuristique + problème” de deux manières différentes (voir figures 5.1 et 5.2).

La cas de la figure 5.1 justifie l'étude des méthodes d'optimisation multiobjectif telle que nous l'avons menée dans les chapitres précédents en ne faisant aucune référence aux métaheuristiques. Le présent chapitre sera consacré aux méthodes d'optimisation multiobjectif indissociables d'une métaheuristique. Nous étudierons principalement des méthodes issues des algorithmes génétiques.

5.3 Généralités

Si l'on considère l'ensemble des méthodes de recherche d'optimum, on constate que celles-ci se réunissent autour de quelques concepts généraux. Par exemple, on trouvera beaucoup de méthodes (méthode de Newton, quasi-Newton, etc.) articulées autour de la descente de gradient. Le concept général de cette famille sera la descente de gradient, et ce concept général sera appelé métaheuristique.

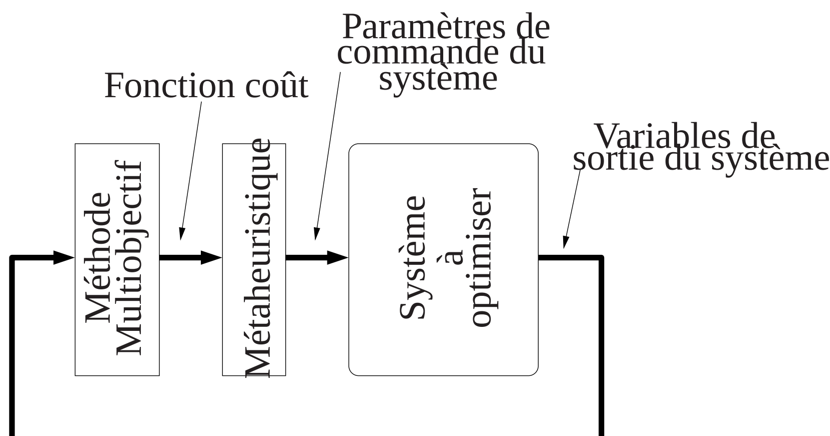


FIG. 5.1 – Décomposition du système lorsqu'il y a découplage.

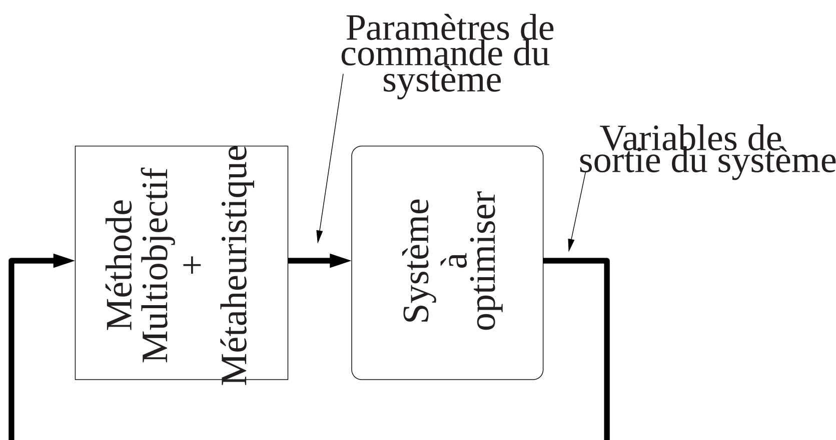


FIG. 5.2 – Décomposition du système lorsqu'il n'y a pas de découplage.

Une métaheuristique est donc une méthode très générale, qui nécessite quelques transformations (mineures en général) avant de pouvoir être appliquée à la résolution d'un problème particulier.

Si l'on considère les différentes métaheuristiques, on constate que celles-ci se répartissent en trois familles :

- Les méthodes déterministes de recherche d'optimum local :
  Ces méthodes convergent rapidement mais, la plupart du temps, elles ne trouvent pas l'optimum global. Elles se contentent de trouver un optimum local et ne reposent pas sur un processus stochastique pour la recherche de l'optimum (voir figure 5.3).
- Les méthodes déterministes de recherche d'optimum global :
  Ces méthodes permettent de trouver un optimum global rapidement et ne reposent pas sur un processus stochastique pour la recherche de l'optimum (voir figure 5.4).
- Les méthodes stochastiques de recherche d'optimum global :
  Ces méthodes reposent sur un processus stochastique chargé d'effectuer la recherche de l'optimum. Elles sont moins performantes (du point de vue rapidité) que les méthodes

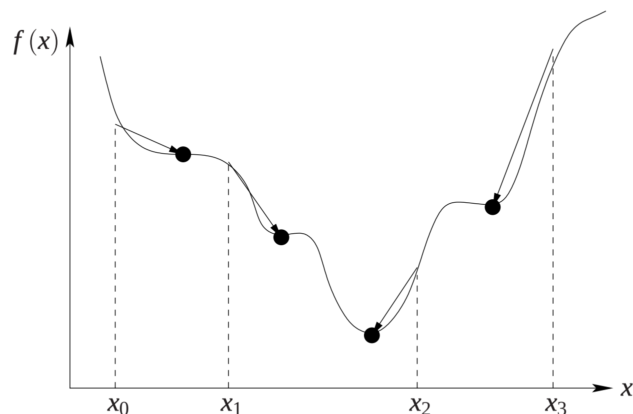


FIG. 5.3 – Méthode déterministe de recherche d'optimum local à partir de différentes solutions.

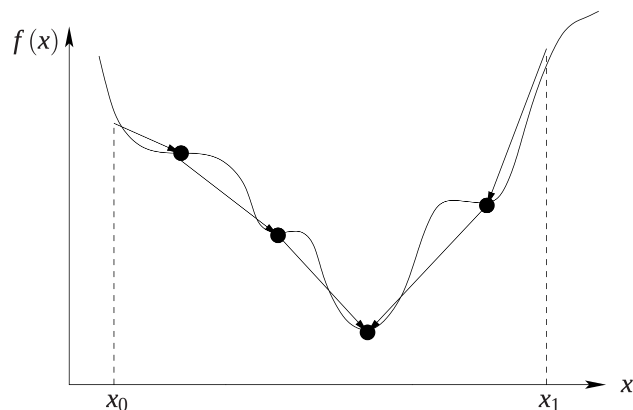


FIG. 5.4 – Méthode déterministe de recherche d'optimum global.

déterministes, mais elles peuvent trouver, en principe, un optimum global difficile à atteindre (voir figure 5.5).

Nous présenterons, dans ce chapitre, des méthodes de recherche appartenant à cette dernière famille.

5.4 Le recuit simulé

Cette méthode de recherche a été proposée par des chercheurs d'IBM qui étudiaient les verres de spin. Ici, on utilise un processus métallurgique (le recuit) pour trouver un minimum. En effet, pour qu'un métal retrouve une structure proche du cristal parfait (l'état cristallin correspond au minimum d'énergie de la structure atomique du métal), on porte celui-ci à une température élevée, puis on le laisse refroidir lentement de manière à ce que les atomes aient le temps de s'ordonner régulièrement (voir [Bonomi 88], [Siarry 94] et [Siarry 99]).

Ce processus métallurgique a été transposé à l'optimisation et a donné une méthode simple et efficace. Le pseudo-code du recuit simulé est représenté dans l'algorithme 5.1.

Le fonctionnement de cet algorithme est le suivant :

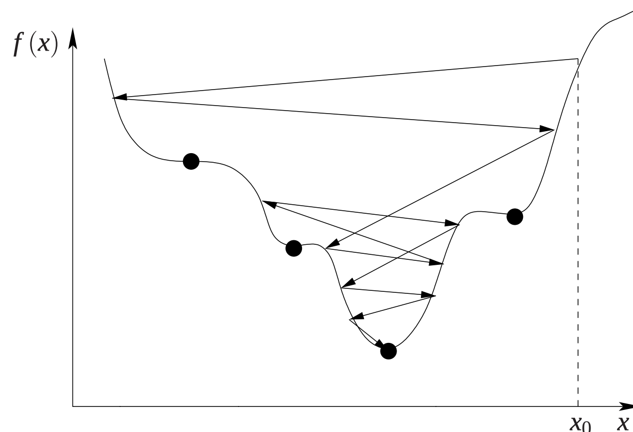


FIG. 5.5 – Méthode stochastique de recherche d'optimum global.

Algorithm 5.1 Recuit simulé.

init T (température initiale)
init x (point de départ)
init ΔT (température minimale)

while(not(end))
  $y = Voisin(x)$
  $\Delta C = C(y) - C(x)$
  if $\Delta C < 0$ then $y = x$
  else if $alea(0,1) < e^{-\frac{\Delta C}{T}}$ then $y = x$
  $T = \quad (T)$
  if $T < \Delta T$ then end(while)
repeat(while)

- On commence par choisir un point de départ au hasard (x).
- On calcule un voisin de ce point ($y = Voisin(x)$).
- On évalue ce point voisin et on calcule l'écart par rapport au point d'origine ($\Delta C = C(y) - C(x)$).
- Si cet écart est négatif, on prend le point y comme nouveau point de départ, s'il est positif, on peut quand même accepter le point y comme nouveau point de départ, mais avec une probabilité $e^{-\frac{\Delta C}{T}}$ (qui varie en sens inverse de la température T).
- Au fur et à mesure du déroulement de l'algorithme, on diminue la température T ($T = \quad (T)$), souvent par paliers.
- On répète toutes ces étapes tant que le système n'est pas figé (par exemple, tant que la température n'a pas atteint un seuil minimal).

Nous allons maintenant décrire deux implémentations du recuit simulé multiobjectif.

5.4.1 La méthode P.A.S.A (*Pareto Archived Simulated Annealing*)

5.4.1.1 Principe

Cette méthode, développée à EDF [Engrand et al. 98], utilise une fonction d'agrégation des fonctions objectif couplée avec un système d'archivage de solutions non dominées.

5.4.1.2 Présentation de la méthode

On suppose que les fonctions objectif du problème d'optimisation multiobjectif sont positives. Cette hypothèse permet d'utiliser la fonction d'agrégation suivante, pour se ramener à un problème de minimisation monobjectif :

$$G\left(\overrightarrow{x}\right) = \bigoplus_{i=1}^{n} \ln\left(f_i\left(\overrightarrow{x}\right)\right) \tag{5.1}$$

De cette manière, l'expression suivante :

$$\Delta G = G\left(\overrightarrow{x}'\right) - G\left(\overrightarrow{x}\right) = \bigoplus_{i=1}^{n} \ln\left(\frac{f_i\left(\overrightarrow{x}'\right)}{f_i\left(\overrightarrow{x}\right)}\right) \tag{5.2}$$

représente la variation relative moyenne des fonctions objectif entre le point courant ($\overrightarrow{x}$) et le point à tester ($\overrightarrow{x}'$) :

- si $\Delta G > 0$, la nouvelle solution $\overrightarrow{x}'$ détériore, en moyenne relative, l'ensemble des fonctions objectif ;
- si $\Delta G < 0$, la nouvelle solution $\overrightarrow{x}'$ améliore, en moyenne relative, l'ensemble des fonctions objectif.

Ensuite, comme pour le recuit simulé original, on définit une probabilité d'acceptation p :

$$p = \exp\left(-\frac{\Delta G}{T}\right) \tag{5.3}$$

où T désigne la température de l'algorithme de recuit. Cette température a la même signification que dans la méthode du recuit simulé monobjectif.

Pour l'archivage des solutions non dominées, on utilise une "archive" de taille variable (entre 0 et N_{max} solutions dans l'archive). La gestion de l'archivage se conforme aux règles suivantes :

- si la solution $\overrightarrow{x}'$ est dominée par au moins une solution de l'archive, alors la solution $\overrightarrow{x}'$ n'est pas archivée ;
- si la solution $\overrightarrow{x}'$ domine une solution $\overrightarrow{y}$ de l'archive, alors la solution $\overrightarrow{x}'$ remplace la solution $\overrightarrow{y}$ dans l'archive ;
- si la solution $\overrightarrow{x}'$ n'est pas dominée par l'archive, alors on place cette solution dans l'archive et on retire de l'archive les solutions qui sont dominées par la solution $\overrightarrow{x}'$.

Certaines autres règles peuvent être ajoutées pour améliorer la gestion des individus de l'archive. Par exemple, on peut ajouter un critère restrictif pour l'ajout d'une solution dans l'archive :

- la solution $\overrightarrow{x}'$ doit dominer toutes les solutions de l'archive et être à une distance minimale de solutions adjacentes de l'archive.

Pour que la recherche se fasse sur toute la surface de compromis, il est nécessaire de relancer régulièrement la recherche à partir d'un point choisi au hasard, au sein de l'archive. Cette étape du choix de l'individu est appelée *return to base*.

L'algorithme de la méthode P.A.S.A est celui décrit dans l'algorithme 5.2.

5.4.1.3 Discussion

Dans cet algorithme, la population initiale présente dans l'archive peut être issue d'une première optimisation multiobjectif. La qualité des solutions initialement présentes dans l'archive influencera la qualité du résultat obtenu à la fin de l'optimisation courante.

Algorithm 5.2 La méthode P.A.S.A

1. Choix de P_0, une population d'archive initiale, éventuellement réduite à un seul élément x_0, et choix de Tq initiale.
2. Soit $\overrightarrow{x_n}'$ un voisin de $\overrightarrow{x_n}$
3. Calcul de $G(\overrightarrow{x_n}')$ et de $\Delta G = G(\overrightarrow{x_n}') - G(\overrightarrow{x_n})$
4. Si $\Delta G < 0$ alors $\overrightarrow{x_{n+1}} = \overrightarrow{x_n}'$
5. Si $\Delta G > 0$ alors soit $p = \exp\left(-\frac{\Delta G}{T_q}\right)$
   - $\overrightarrow{x_{n+1}} = \overrightarrow{x_n}'$ avec une probabilité p
   - $\overrightarrow{x_{n+1}} = \overrightarrow{x_n}$ avec une probabilité $1 - p$
6. Si on sort de la boucle interne, alors $T_{q+1} = \quad \cdot T_q$ avec $\quad < 1$
7. Principe de non-dominance pour l'archivage de $\overrightarrow{x_{n+1}}$
   - si $\overrightarrow{x_{n+1}}$ domine une des solutions archivées $\overrightarrow{y_p}$, $\overrightarrow{x_{n+1}}$ remplace $\overrightarrow{y_p}$ dans la population archivée ;
   - si $\overrightarrow{x_{n+1}}$ est dominée par au moins une des solutions archivées, alors on ne l'archive pas ;
   - si $\overrightarrow{x_{n+1}}$ n'est pas dominée par la population archivée alors on l'ajoute à la population archivée, et on retire de celle-ci les solutions dominées par $\overrightarrow{x_{n+1}}$.
8. *Return to base* : lorsque c'est nécessaire, $\overrightarrow{x_{n+1}} = \overrightarrow{y_i}$ (i choisi aléatoirement dans la population archivée)
9. Si (non convergence) alors retour en 2.

Un autre avantage de cette méthode est la facilité avec laquelle on peut inclure des règles heuristiques pour la gestion de l'archive. Cette “flexibilité” peut permettre d'intégrer un “savoir ingénieur”, qui est susceptible d'améliorer la qualité de la population obtenue à la fin de l'optimisation, mais surtout la vitesse de convergence vers cette population d'individus archivés.

Un défaut de cette méthode que l'on peut souligner est que la forme d'agrégation de la fonction objectif (l'équation 5.1) utilisée ne supporte que les valeurs positives de fonctions objectif.

5.4.2 La méthode M.O.S.A (*Multiple Objective Simulated Annealing*)

5.4.2.1 Principe

Cette méthode a été proposée dans [Ulungu et al. 99]. Elle utilise le recuit simulé pour rechercher la surface de compromis.

5.4.2.2 Présentation de la méthode

On commence par définir une suite de fonctions donnant la probabilité d'acceptation d'une mauvaise solution, pour chaque fonction objectif :

$$\Pi_k = \begin{cases} \exp\left(-\frac{\Delta f_k}{T_n}\right) & \text{si } \Delta f_k > 0 \\ 1 & \text{si } \Delta f_k \leq 0 \end{cases} \tag{5.4}$$

avec

T_n	la température du recuit simulé à l'itération n,
f_k	k^{me} fonction objectif,
$\overrightarrow{x_n}$	solution obtenue à l'itération n,
$\overrightarrow{y}$	point considéré lors de l'itération n,

$\Delta f_k = f_k(\overrightarrow{y}) - f_k(\overrightarrow{x}_n)$,

Ensuite, une fois que toutes ces probabilités ont été calculées, on les agrège. Pour cela, il existe plusieurs solutions :

– on effectue le produit des probabilités :

$$t(\Pi, \quad) = \prod_{k=1}^{N} (\Pi_k)^{\quad k} \tag{5.5}$$

avec

Π	ensemble des Π_k, $k = \{1, \cdots, N\}$
	ensemble des $\quad_k$, $k = \{1, \cdots, N\}$

– on prend la plus petite probabilité :

$$t(\Pi, \quad) = \min_{k=1,\cdots,N} (\Pi_k)^{\quad k} \tag{5.6}$$

Ici, $\quad_k$ correspond à un coefficient de pondération relatif à une fonction objectif. Ce coefficient permet de prendre en compte un certain ordre entre les différents objectifs.

Pour finir, il y a la phase de sélection de la solution. Cette phase s'effectue avec une condition d'acceptation d'une solution, qui est alors la suivante :

Tout d'abord, on a :

$$f_{eq}(\overrightarrow{x}) = \bigoplus_{i=1}^{N} w_i \cdot f_i(\overrightarrow{x}) \tag{5.7}$$

et $\Delta f_{eq} = f_{eq}(\overrightarrow{x_{n+1}}) - f_{eq}(\overrightarrow{x_n})$

– si $\Delta f_{eq} \leq 0$ alors $\overrightarrow{x_{n+1}} = \overrightarrow{y}$
– si $\Delta f_{eq} > 0$ alors :
  $\overrightarrow{x_{n+1}} = \overrightarrow{y}$ avec la probabilité p
  $\overrightarrow{x_{n+1}} = \overrightarrow{x_n}$ avec la probabilité $1 - p$

Cette méthode de traitement multiobjectif peut paraître semblable à la méthode de pondération des fonctions objectif ; cependant, une différence importante existe :

> dans la méthode de pondération des fonctions objectif, on commence par effectuer une somme pondérée des fonctions objectif (voir équation 5.7).

Lorsque cette somme pondérée est "traitée" par la méthode du recuit simulé, elle est prise en compte sous la forme d'une exponentielle :

$$p = \exp\left(-\frac{\Delta f_{eq}}{T_n}\right) \tag{5.8}$$

Cette probabilité d'acceptation peut alors se décomposer de la manière suivante :

$$p = \exp\left(-\frac{\bigoplus_{i=1}^{N} w_i \cdot (f_i(\overrightarrow{x}) - f_i(\overrightarrow{x}_n^*))}{T_n}\right) \tag{5.9}$$

$$p = \prod_{i=1}^{N} \exp\left(-\frac{w_i \cdot (f_i(\overrightarrow{x}) - f_i(\overrightarrow{x}_n^*))}{T_n}\right) \tag{5.10}$$

$$p = \prod_{i=1}^{N} \left(\exp\left(-\frac{\Delta f_i}{T_n}\right)\right)^{w_i} \tag{5.11}$$

$$p = \prod_{i=1}^{N} \Pi_i^{w_i} \tag{5.12}$$

La condition d'acceptation d'une solution est alors la suivante :

– si $\Delta f_{eq} \leq 0$ alors $\overrightarrow{x_{n+1}} = \overrightarrow{y}$
– si $\Delta f_{eq} > 0$ alors :
  $\overrightarrow{x_{n+1}} = \overrightarrow{y}$ avec la probabilité p
  $\overrightarrow{x_{n+1}} = \overrightarrow{x_n}$ avec la probabilité $1 - p$
– la différence avec la méthode MOSA réside dans le calcul de la probabilité d'acceptation.
  Dans la méthode MOSA, on commence par calculer des probabilités par fonction objectif :

$$p_i = \exp\left(-\frac{\Delta f_i}{T_n}\right)$$

  alors qu'avec la méthode de pondération des fonctions objectif, le calcul de la probabilité s'effectue sur la fonction objectif équivalente :

$$p = \exp\left(-\frac{\Delta f_{eq}}{T_n}\right)$$

La méthode MOSA prend plus en considération l'aspect multiobjectif du problème d'optimisation que la méthode de pondération des fonctions objectif traitée par la méthode du recuit simulé.

5.4.2.3 Discussion

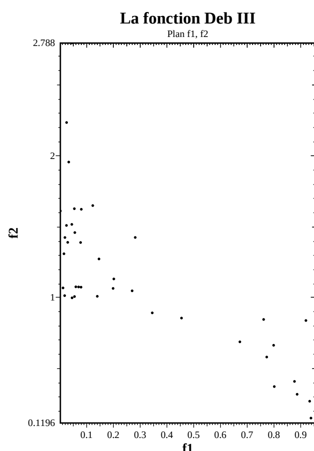
Dans [Ulungu et al. 99], la méthode MOSA est utilisée de la manière suivante, pour un problème d'optimisation combinatoire biobjectif.

1. Deux couples de coefficients de pondération sont calculés.
2. Pour chaque couple, une population de solutions est calculée. Chaque couple va "remplir" une partie de la surface de compromis.
3. Les deux populations sont alors réunies, puis les individus non dominants de la population totale sont éliminés.

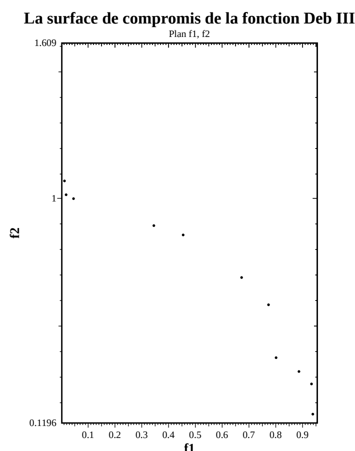
Cette méthode fonctionne bien car, à haute température, le recuit simulé répartit les individus sur toute une surface, et non sur un point. A la fin de l'optimisation, la température de fonctionnement est encore suffisamment élevée pour qu'un "effet stochastique" distribue les solutions potentielles sur un front.

Cet effet peut être souligné par l'exemple suivant, qui concerne un problème test biobjectif non convexe, le problème test de Deb :

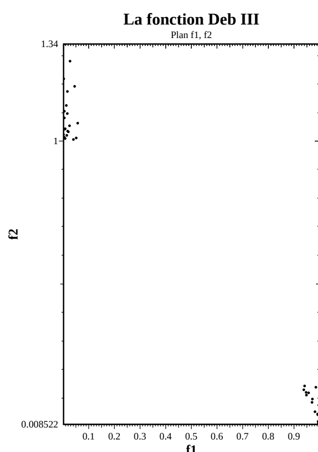
1. On calcule 40 couples de coefficients de pondération,
2. Pour chaque couple, une solution au problème monobjectif équivalent est recherchée, en utilisant la méthode du recuit simulé,



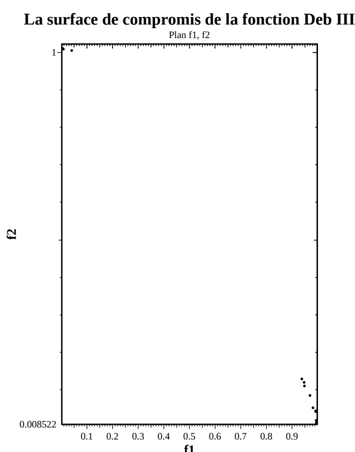
(a) L'ensemble obtenu après 10^5 itérations



(b) La surface de compromis après 10^5 itérations



(c) L'ensemble obtenu après 10^6 itérations



(d) La surface de compromis après 10^6 itérations

FIG. 5.6 – Influence de l'effet stochastique sur la qualité de la surface de compromis.

3. Cette séquence est appliquée au problème test pendant 10^5 itérations, puis 10^6 itérations.

Comme on peut le remarquer à la figure 5.6, la recherche de la surface de compromis avec la méthode de pondération des fonctions objectif couplée avec la méthode du recuit simulé pendant 10^5 itérations produit toute la surface de compromis, alors que cette méthode est censée être inefficace pour ce type de problème.

Ensuite, pour 10^6 itérations, on voit que les solutions se répartissent autour de deux points : c'est l'enveloppe convexe du problème test non convexe.

Donc, comme le montre cette figure, l'effet stochastique a permis de trouver des solutions dans des régions "non accessibles" pour la méthode de pondération des fonctions objectif.

5.5 La recherche Tabou

Cette méthode, mise au point par F. Glover [Glover 86, Glover 90, Glover et al. 97], est conçue en vue de surmonter les minima locaux de la fonction objectif. C'est une technique d'optimisation combinatoire que certains présentent comme une alternative au recuit simulé.

A partir d'une configuration initiale quelconque, Tabou engendre une succession de configurations qui doit aboutir à la configuration optimale. A chaque itération, le mécanisme de passage d'une configuration, soit x_n, à la suivante, soit x_{n+1}, est le suivant :

– on construit l'ensemble des "voisins" de x_n, c'est-à-dire l'ensemble des configurations accessibles en un seul "mouvement" élémentaire à partir de x_n (si cet ensemble est trop vaste, on en extrait aléatoirement un sous-ensemble de taille fixée) : soit $\mathit{Voisinage}(x_n)$ l'ensemble (ou le sous-ensemble) envisagé ;
– on évalue la fonction objectif f du problème pour chacune des configurations appartenant à $\mathit{Voisinage}(x_n)$. La configuration x_{n+1}, qui succède à la configuration x_n dans la chaîne de Markov construite par Tabou, est la configuration de $\mathit{Voisinage}(x_n)$ en laquelle f prend sa valeur minimale.

Notons que la configuration x_{n+1} est adoptée même si $f(x_{n+1}) > f(x_n)$: c'est grâce à cette particularité que Tabou permet d'éviter les minima locaux de f.

Cependant, telle quelle la procédure ne fonctionne généralement pas, car il y a un risque important de retourner à une configuration déjà retenue lors d'une itération précédente, ce qui provoque l'apparition d'un cycle. Pour éviter ce phénomène, on tient à jour, à chaque itération, une "liste Tabou" de mouvements interdits ; cette liste - qui a donné son nom à la méthode - contient les mouvements inverses ($x_{n+1} \rightarrow x_n$) des m derniers mouvements ($x_n \rightarrow x_{n+1}$) effectués (typiquement $m = 7$). La recherche du successeur de la configuration courante x_n est alors restreinte aux voisins de x_n qui peuvent être atteints sans utiliser un mouvement de la liste tabou. La procédure peut être stoppée dès que l'on a effectué un nombre donné d'itérations, sans améliorer la meilleure solution atteinte jusqu'ici.

L'algorithme ainsi décrit est dit "Tabou simple". Selon [De Werra 90], il serait plus efficace que le recuit simulé pour le problème modèle du "coloriage d'un graphe". Cependant, le mode de construction de la liste tabou - qui, pour une simple raison d'économie de place mémoire, contient des mouvements interdits, et non des configurations interdites - peut bloquer l'accès à certaines solutions, pourtant non encore visitées. Pour éviter cet inconvénient, on peut employer la méthode plus complexe dite "Tabou généralisée", qui prévoit la possibilité d'annuler le statut tabou d'un mouvement, lorsque le bénéfice escompté est "suffisant" (cette circonstance est appréciée à l'aide de la notion de "niveau d'aspiration", qui est précisée en détail dans les articles référencés).

Le pseudo-code de cette méthode est représenté dans l'algorithme 5.3.

5.6 Les algorithmes génétiques

Les algorithmes génétiques sont inspirés de la génétique classique (voir [Goldberg 94]) : on considère une "population" de points répartis dans l'espace.

Algorithm 5.3 Recherche Tabou.

init $L = \emptyset$ (liste des points tabous)
init x (point de départ)
init k_{fin} (nb d'itérations max total)
init $k = 0$ et $l = 0$
while(not(end))
 $y = Voisinage(x) - L$ (on prend un voisinage sans points tabous)
 $\Delta C = C(y) - C(x)$
 $x = y$ avec $y \in Voisinage(x) - L$ tel que $\min C(y) - C(x)$
 $L = L + \{x\}$
 $k = k + 1$
 Update L (on retire des mouvements tabou de L suivant des critères d'aspiration)
 if $k = k_{fin}$ then end(while)
repeat

Avant d'expliquer en détail le fonctionnement d'un algorithme génétique, nous allons présenter quelques mots de vocabulaire relatifs à la génétique. Ces mots sont souvent utilisés pour décrire un algorithme génétique.

Génotype ou **chromosome :** c'est une autre façon de dire "individu".

Individu : correspond au codage sous forme de gènes d'une solution potentielle à un problème d'optimisation.

Gène : un chromosome est composé de gènes. Dans le codage binaire, un gène vaut soit 0 soit 1.

Phénotype : chaque génotype représente une solution potentielle à un problème d'optimisation. La valeur de cette solution potentielle est appelée le phénotype.

Tout ce vocabulaire est illustré à la figure 5.7.

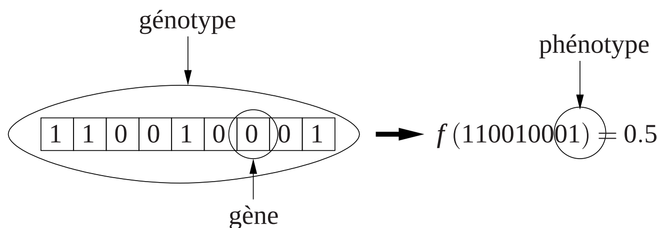


FIG. 5.7 – Le vocabulaire des algorithmes génétiques.

– Chaque individu va être "codé" à la manière d'un gène. Par exemple, le plus souvent, on fait correspondre une chaîne binaire à l'individu (cette chaîne binaire est une image de la position de l'individu dans l'espace de recherche). Par exemple, trois individus (de 8 bits chacun) sont représentés à la figure 5.8.

| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

(a) Individu 1

| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

(b) Individu 2

| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |

(c) Individu 3

FIG. 5.8 – Trois individus.

– A chaque individu on associe une efficacité (on appelle aussi cette valeur l' "adaptation"). Cette efficacité va correspondre à la performance d'un individu dans la résolution d'un problème donné. Par exemple, si l'on considère un problème de maximisation d'une fonction, l'efficacité de l'individu croîtra avec sa capacité à maximiser cette fonction.
  Si l'on prend l'exemple de la maximisation d'une fonction $f(x)$, on a le tableau 5.1.

$f(x)$	10	20	5
Efficacité	2	4	1
Individu	1	2	3

TAB. 5.1 – Un exemple de valeurs d'efficacité.

  Comme on peut le voir sur cet exemple, l'individu 1 est moins efficace que l'individu 2. En effet, il maximise moins $f(x)$ que l'individu 2.
– Après avoir déterminé l'efficacité des individus, on opère une reproduction. On copie les individus proportionnellement à leur efficacité. Par exemple, si l'individu 1 est reproduit deux fois, l'individu 2 sera reproduit quatre fois et l'individu 3 est reproduit une fois.
  Ce schéma de fonctionnement est à l'image de la vie réelle. Plus un individu est adapté à son milieu (il se défend mieux contre ses prédateurs, il mange mieux) plus sa capacité à se reproduire sera grande.
– Une fois que l'on a déterminé la fonction d'efficacité et opéré la reproduction, on effectue des "croisements" entre individus (définis plus loin). Plus un individu sera efficace dans la résolution du problème, plus celui-ci se croisera avec un grand nombre d'autres individus. Pour cela, on utilise une roue sur laquelle un individu occupe une place qui est proportionnelle à son efficacité (*roulette wheel selection* en anglais). Cette roue, dans le cas de l'exemple que l'on a traité ci-dessus, est représentée à la figure 5.9.

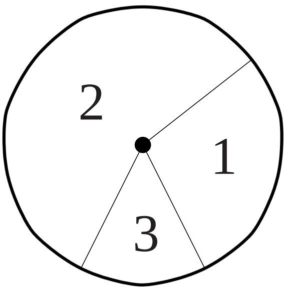


FIG. 5.9 – Roue servant à sélectionner un individu pour le croisement.

  Pour notre exemple, on aura le tableau 5.2.
– On va maintenant fusionner l'individu de départ avec l'individu associé. Cette opération s'appelle le croisement. Pour cela, on sélectionne aléatoirement une position dans le codage de l'individu. A cette position, on sépare le codage puis on échange les morceaux de droite entre les deux individus. Un exemple de croisement est représenté à la figure 5.10.
  On peut répéter cette opération pour tous les individus du tableau précédent. On obtient alors le tableau 5.3."

Sélection pour le croisement	
"Mâle"	"Femelle"
Individu sélectionné suivant l'efficacité	Individu sélectionné aléatoirement
1	2
1	3
2	2
2	2
2	3
2	1
3	2

TAB. 5.2 – Le croisement, première étape.

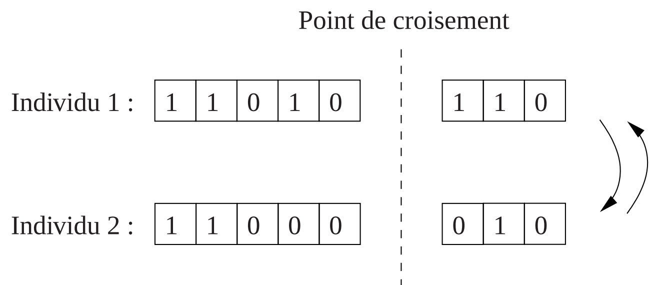


FIG. 5.10 – Le croisement, deuxième étape.

"Mâle"	"Femelle"	Position de croisement	Résultat
1 1 0 1 0 1 1 0 1	1 0 1 0 1 1 0 0 2	1	1 0 1 0 1 1 0 0 1 1 0 1 0 1 1 0
1 1 0 1 0 1 1 0 1	1 1 0 0 0 0 1 0 3	3	1 1 0 0 0 0 1 0 1 1 0 1 0 1 1 0
1 0 1 0 1 1 0 0 2	1 0 1 0 1 1 0 0 2	2	1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0
1 0 1 0 1 1 0 0 2	1 0 1 0 1 1 0 0 2	5	1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 0
1 0 1 0 1 1 0 0 2	1 1 0 0 0 0 1 0 3	7	1 0 1 0 1 1 0 0 1 1 0 0 0 0 1 0
1 0 1 0 1 1 0 0 3	1 1 0 1 0 1 1 0 1	4	1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 0
1 1 0 0 0 0 1 0 3	1 0 1 0 1 1 0 0 2	4	1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 0

TAB. 5.3 – Le croisement, résultat final.

Ici, on constate que l'on a obtenu deux nouveaux éléments dans la colonne résultat. Notre population s'est diversifiée. A ce moment, soit on conserve toute la population (mais on supprime les doublons), soit on supprime les éléments les moins performants, de manière à conserver le même nombre d'individus dans la population. D'autres méthodes de réduction

de population peuvent être mises en œuvre ici (par exemple, on sélectionne au hasard les individus qui vont être supprimés).

- Pour finir, on peut procéder à une mutation au niveau des gènes d'un individu.
  Pour cela, on commence par choisir au hasard un certain nombre d'individus (en général, la probabilité de sélection pour la mutation est faible).
  Ensuite, pour chaque individu sélectionné, on choisit au hasard le gène où aura lieu la mutation.
  Finalement, à cette position, on change le 0 en 1 et réciproquement.
- On recommence ce procédé jusqu'à ce que l'on atteigne un critère d'arrêt (par exemple, un nombre maximal d'itérations).

Le pseudo-code d'un algorithme génétique est donné dans l'algorithme 5.4.

Algorithm 5.4 Un algorithme génétique.

```
Initialisation de la population
Evaluation des fonctions objectif
Calcul de l'efficacité
For i=1 to MaxIter
  Sélection aléatoire,
  Sélection proportionnelle à l'efficacité
  Croisement
  Mutation
  Evaluation des fonctions objectif
  Calcul de l'efficacité
End For
```

L'avantage de cette méthode est que, dans certains cas, elle peut procurer un jeu de solutions optimales.

5.7 L'optimisation multiobjectif et les algorithmes génétiques

Les algorithmes génétiques sont très bien adaptés au traitement d'un problème d'optimisation multiobjectif. En témoigne le nombre important d'articles qui ont été publiés sur ce sujet. De plus, ce domaine est très dynamique et ne cesse de se développer.

Dans le paragraphe précédent, nous avons vu l'algorithme génétique. Le schéma de fonctionnement d'un algorithme génétique pour la résolution d'un problème d'optimisation monobjectif est représenté à la figure 5.11.

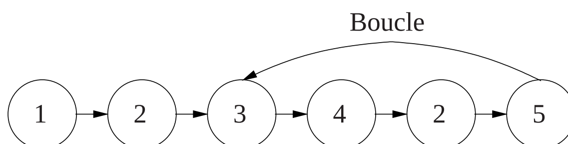


FIG. 5.11 – Le fonctionnement d'un algorithme génétique monobjectif.

1	Initialisation de la population.
2	Evaluation de l'efficacité des individus de la population.
3	Croisement.

4 Mutation.

5 Sélection.

Le fonctionnement d'un algorithme génétique traitant un problème d'optimisation multiobjectif est représenté à la figure 5.12.

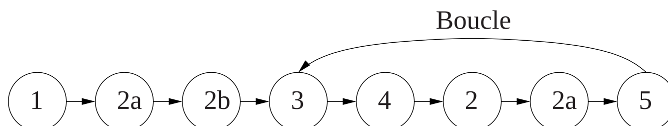


FIG. 5.12 – Le fonctionnement d'un algorithme génétique multiobjectif.

1 Initialisation de la population.

2a Evaluation de l'efficacité des individus de la population.

2b Transformation vecteur/efficacité.

3 Croisement.

4 Mutation.

5 Sélection.

La seule différence que l'on note est l'étape 2b. A cette étape, on transforme un vecteur (qui contient les valeurs des fonctions objectif pour chaque individu) en une efficacité. Diverses stratégies, qui seront décrites dans les paragraphes suivants, peuvent être adoptées.

Tous les algorithmes génétiques traitant un problème d'optimisation multiobjectif pourront se ramener à ce schéma.

Nous allons maintenant nous intéresser aux diverses méthodes qui ont été mises en œuvre pour le traitement de problèmes d'optimisation multiobjectif. Ces méthodes se décomposeront en deux familles :

- Une première famille comportera les méthodes dites “non agrégatives”. Dans ces méthodes, il n'y a pas fusion des différentes fonctions objectif pour se ramener à un problème d'optimisation monobjectif.
- La deuxième famille comportera les méthodes “agrégatives”. Le but de ces méthodes est de se ramener à un problème d'optimisation monobjectif en utilisant une fonction objectif équivalente.

Note : La plupart des algorithmes utilisent une “distance” qui caractérise la différence entre deux individus. La distance entre l'individu i et l'individu j est notée $d(i,j)$. On calcule cette distance de la manière suivante : chaque différence entre les motifs (gènes) des individus compte pour $+1$ (voir figure 5.13) : c'est la distance de Hamming.

Pour cet exemple, on constate qu'il y a cinq différences entre l'individu 1 et l'individu 2. Par conséquent, la distance entre l'individu 1 et l'individu 2 vaut :

$$d(1,2) = 5 \tag{5.13}$$

5.7.1 Les méthodes “non agrégatives”

5.7.1.1 La méthode *Vector Evaluated Genetic Algorithm* (V.E.G.A.)

5.7.1.1.1 Principe

Cette méthode permet de traiter un problème d'optimisation multiobjectif sans avoir à agréger les fonctions objectif en une seule fonction [Coello 98, Deb 99].

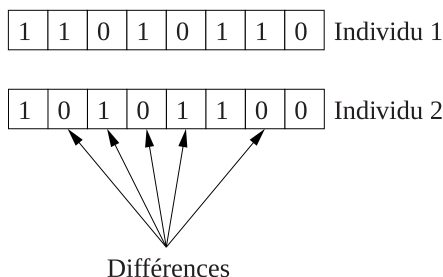


FIG. 5.13 – La distance de Hamming.

5.7.1.1.2 Présentation de la méthode

L'algorithme V.E.G.A. considère une population de N individus. Ces N individus sont répartis en k groupes (k étant le nombre de fonctions objectif de notre problème) de $\frac{N}{k}$ individus (avec N multiple de k). A chaque groupe, on associe une fonction objectif. Cette fonction objectif permet de déterminer l'efficacité d'un individu au sein du groupe. Ensuite, les individus sont mélangés et les croisements sont opérés en tenant compte de l'efficacité de chaque individu.

La figure 5.14 représente les différentes séquences de fonctionnement de la méthode. Sur cette figure, on a représenté les différents types de populations que l'on traite au cours du déroulement de la méthode V.E.G.A. (soit un ensemble d'individus, soit des groupes d'individus).

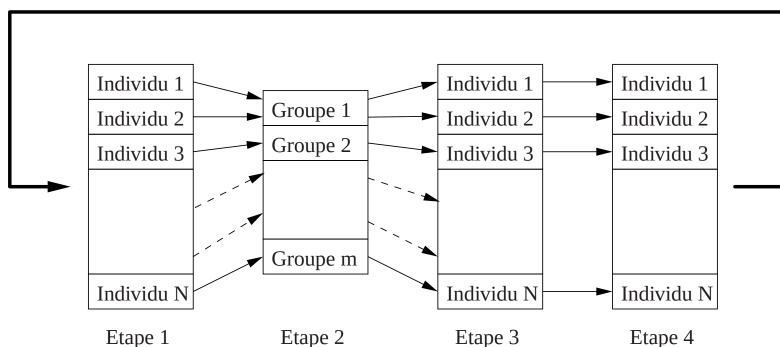


FIG. 5.14 – Principe de l'algorithme V.E.G.A.

Voici la description des différentes étapes :

Etape 1 : Itération i. Initialisation d'une population de taille N.

Etape 2 : Création des k groupes (sous-populations).

Etape 3 : Calcul des efficacités.
Mélange des individus.

Etape 4 : On applique l'algorithme génétique classique (croisement-mutation- sélection).
Puis on passe à l'itération suivante ($i+1$).

5.7.1.1.3 Discussion

Le danger de cette méthode est d'obtenir, en fin d'optimisation, une population constituée

d'individus moyens dans tous les objectifs. Une telle population ne permet pas d'obtenir une surface de compromis bien dessinée. En effet, la population va se concentrer autour d'un "point" moyen. Des artifices ont été trouvés pour contrebalancer cet effet (utilisation d'espèce au sein d'un groupe, etc.).

De plus, il a été montré que cette méthode est équivalente à la méthode de pondération des fonctions objectif. Donc, elle ne permet pas de trouver des solutions qui se trouveraient dans une concavité.

5.7.2 Les méthodes "agrégatives"

5.7.2.1 La méthode *Multiple Objective Genetic Algorithm* (M.O.G.A.)

5.7.2.1.1 Principe

Cette méthode est présentée dans [Deb 99] et [Fonseca et al 93]. Elle utilise la relation de dominance pour déterminer l'efficacité d'un individu.

5.7.2.1.2 Présentation de la méthode

Cette méthode est basée sur la dominance au sens de Pareto. Ici, le "rang" d'un individu (numéro d'ordre qui permet de classer un individu par rapport aux autres) est donné par le nombre d'individus qui dominent l'individu considéré.

Par exemple, si l'on considère un individu x_i à la génération t qui est dominé par $p_i^{(t)}$ individus, le rang de l'individu considéré est donné par :

$$rang\,(x_i, t) = 1 + p_i^{(t)} \tag{5.14}$$

A tous les individus non dominés on affecte le rang 1. Les individus dominés se voient donc affectés à un rang important (une forte pénalité donc).

Pour le calcul de l'efficacité, on peut suivre les étapes suivantes :

1. Classer les individus en fonction de leur rang.
2. Affecter une efficacité à un individu en interpolant à partir du meilleur (rang 1) jusqu'au plus mauvais (rang n), en utilisant une fonction $f\,(rang)$. Cette fonction est, le plus souvent, linéaire (voir figure 5.15).

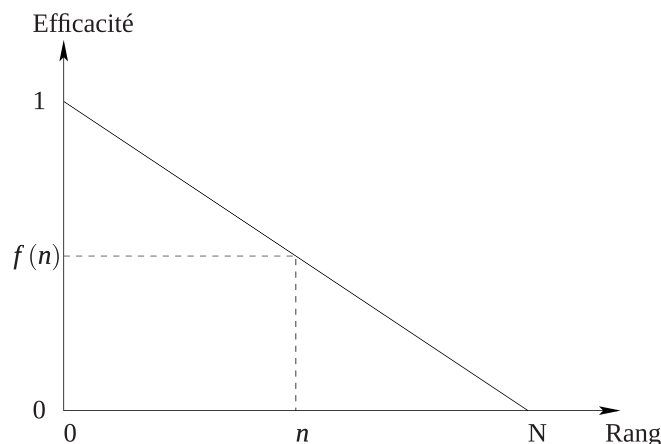


FIG. 5.15 – Un exemple de fonction d'efficacité.

Le pseudo-code de cette méthode est représenté dans l'algorithme 5.5.

Algorithm 5.5 Algorithme MOGA.

Initialisation de la population
Evaluation des fonctions objectif
Assignation d'un rang basé sur la dominance
Assignation d'une efficacité à partir du rang
For i=1 to G
  Sélection aléatoire proportionnelle à l'efficacité
  Croisement
  Mutation
  Evaluation des fonctions objectif
  Assignation d'un rang basé sur la dominance
  Assignation d'une efficacité à partir du rang
End For

5.7.2.1.3 Discussion

La méthode MOGA ne permet pas, dans certains cas, d'obtenir une diversité dans la représentation des solutions. Les méthodes N.S.G.A et N.P.G.A, qui vont être présentées maintenant, permettent d'éviter cette difficulté.

Pour ce type de méthode, le nombre de comparaisons effectuées par génération s'écrit :

$$N \cdot (N-1) \cdot n \tag{5.15}$$

où N correspond au nombre de points dans la population et n au nombre de fonctions objectif.

5.7.2.2 La méthode *Non dominated Sorting Genetic Algorithm* (N.S.G.A.)

5.7.2.2.1 Principe

Cette méthode reprend les grandes lignes de la méthode M.O.G.A précédemment décrite. La différence principale intervient lors du calcul de l'efficacité d'un individu [Srinivas et al. 93].

5.7.2.2.2 Présentation de la méthode

Cette méthode est basée sur une classification en plusieurs niveaux des individus. Dans un premier temps, avant de procéder à la sélection, on affecte à chaque individu de la population un rang (en utilisant le rang de Pareto, voir section 1.6.2). Tous les individus non dominés de même rang sont classés dans une catégorie. A cette catégorie, on affecte une efficacité factice, qui est inversement proportionnelle au rang de Pareto de la catégorie considérée.

On veut maintenant que les individus de la catégorie considérée se répartissent de manière uniforme au sein de celle-ci. On veut donc une bonne représentation (voir section 1.11), ou encore on veut qu'il y ait une diversité de solutions.

Pour maintenir la diversité de la population, ces individus classés se voient affectés d'une nouvelle valeur d'efficacité. Pour cela, on utilise la formule suivante :

$$m_i = \bigoplus_{j=1}^{K} Sh\left(d\left(i,j\right)\right) \tag{5.16}$$

$$Sh\left(d\left(i,\, j\right)\right) = \begin{cases} 1 - \left(\frac{d(i,j)}{share}\right)^2 & \text{si } d\left(i,j\right) < \quad share \\ 0 & \text{sinon} \end{cases} \tag{5.17}$$

Ici, K désigne le nombre d’individus dans la catégorie considérée et $d(i,j)$ est la distance entre l’individu i et l’individu j telle qu’elle a été introduite dans la section 5.7. Il est toujours possible d’utiliser la distance classique à la place de cette dernière. $_{share}$ est la distance d’influence. Comme on peut le voir dans l’expression ci-dessus, tous les individus qui sont suffisamment proches (dont la distance $d(i,j)$ est inférieure à $_{share}$) seront pris en compte dans le calcul de m_i. Les autres seront ignorés.

La valeur d’efficacité de l’individu i au sein de la catégorie considérée sera alors :

$$f_i = \frac{F}{m_i} \tag{5.18}$$

où F est la valeur d’efficacité affectée à la catégorie à laquelle appartient l’individu. La figure 5.16 montre les différents paramètres qui interviennent dans le calcul de l’efficacité d’un individu.

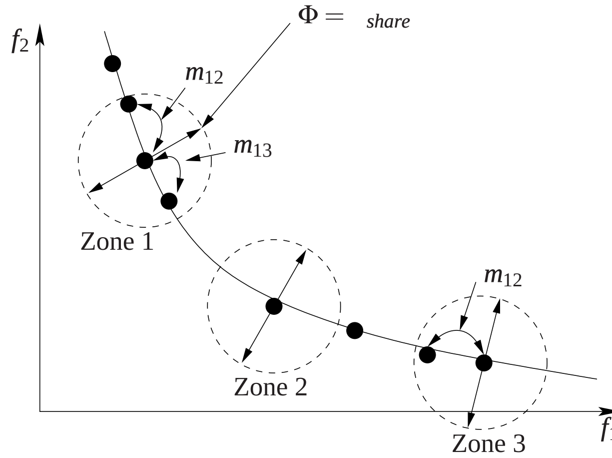


FIG. 5.16 – Le calcul de l’efficacité.

Comme on voit à la figure 5.16, le paramètre $_{share}$ permet de définir une zone d’influence pour le calcul de l’efficacité d’un individu.

Ensuite, on ignore les individus de ce groupe. Le processus reprend alors avec les individus restants, où l’on opère une nouvelle classification, une nouvelle catégorisation et une nouvelle opération de partage. Ce processus est répété jusqu’à ce que tous les individus aient été traités.

Comme les individus qui ont un rang de Pareto de valeur 1 ont une meilleure efficacité, ils sont reproduits en plus grand nombre que les autres. Cette propriété permet d’obtenir une convergence plus rapide vers la surface de compromis. Le partage, lui, permet de maintenir une répartition “uniforme” sur la surface de compromis.

Le pseudo-code de cette méthode est donné dans l’algorithme 5.6.

On distingue deux types de niches :

– les niches génotypiques :

$$Sh_{gen}(d(i,j)) = \begin{cases} 1 - \left(\frac{d_{ham}(i,j)}{\ _{gen}}\right)^2 & \text{si } d_{hamming}(i,j) < \ _{gen} \\ 0 & \text{sinon} \end{cases} \tag{5.19}$$

on effectue le comptage de niches (de rayon $_{gen}$) dans l’espace des gènes, la distance que l’on considère est la distance de Hamming ($d_{ham}(i,j)$)

Algorithm 5.6 Algorithme NSGA.

Initialisation de la population
Evaluation des fonctions objectif
Assignation d'un rang basée sur le rang de dominance sur
  chaque surface de compromis
Calcul du compte des voisins
Assignation d'une efficacité partagée
For i=1 to G
  Sélection aléatoire proportionnelle à l'efficacité
  Croisement
  Mutation
  Evaluation des fonctions objectif
  Assignation d'un rang basée sur le rang de dominance sur
    chaque surface de compromis
  Calcul du compte des voisins
  Assignation d'une efficacité partagée
End For

– les niches phénotypiques :

$$Sh_{phen}\left(d\left(i,j\right)\right)=\begin{cases}1-\left(\frac{d_{euclid}(i,j)}{{}_{phen}}\right)^{2} & \text{si } d_{euclid}\left(i,j\right)< {}_{phen}\\ 0 & \text{sinon}\end{cases} \tag{5.20}$$

on effectue le comptage de niches (de rayon ${}_{phen}$) dans l'espace des phénotypes, la distance que l'on considère est la distance Euclidienne (si l'on considère un problème d'optimisation continu)

– les niches combinées :

$$Sh_{comb}\left(d\left(i,j\right)\right)=\begin{cases}1-\left(\frac{d_{ham}(i,j)}{{}_{gen}}\cdot\frac{d_{euclid}(i,j)}{{}_{phen}}\right)^{2} & \text{si } d_{euclid}\left(i,j\right)< {}_{phen} \\ & \text{et } d_{ham}\left(i,j\right)< {}_{gen}\\ 1-\left(\frac{d_{euclid}(i,j)}{{}_{phen}}\right)^{2} & \text{si } d_{euclid}\left(i,j\right)< {}_{phen} \\ & \text{et } d_{ham}\left(i,j\right)\geq {}_{gen}\\ 1-\left(\frac{d_{ham}(i,j)}{{}_{gen}}\right)^{2} & \text{si } d_{euclid}\left(i,j\right)\geq {}_{phen} \\ & \text{et } d_{ham}\left(i,j\right)< {}_{gen}\\ 0 & \text{sinon}\end{cases}$$

on effectue le comptage de niches (de rayons ${}_{gen}$ et ${}_{phen}$) dans l'espace des génotypes et des phénotypes.

Une niche phénotypique permet de "régler" la diversité des solutions dans l'espace des fonctions objectif alors que la niche génotypique permet de régler la diversité des solutions dans l'espace de décision.

La niche combinée, elle, permet d'obtenir des solutions diversifiées aussi bien dans l'espace de décision que dans l'espace des fonctions objectif.

5.7.2.2.3 Discussion

L'efficacité de cette méthode tient dans le fait que les objectifs sont réduits à une valeur d'efficacité factice obtenue en utilisant le classement en fonction du rang de Pareto. Cette méthode présente l'inconvénient d'être sensible au choix de la valeur de ${}_{share}$.

Pour cette méthode, le nombre de comparaisons effectuées sur une population pour une génération est le même que pour la méthode MOGA. Cependant, un surplus de calculs dû au partage apparaît. Ce surplus est proportionnel à $N \cdot (N-1)$.

5.7.2.3 La méthode *Niched Pareto G.A.* (N.P.G.A.)

5.7.2.3.1 Principe

Cette méthode reprend les grandes lignes de la méthode N.S.G.A précédemment décrite. La différence principale intervient lors du processus de sélection [Horn et al. 93].

5.7.2.3.2 Présentation de la méthode

Dans l'algorithme génétique classique, la méthode de sélection entre deux individus utilise la roue de sélection telle qu'elle a été décrite dans la section 5.6. Dans cette méthode, seule la manière de sélectionner les individus change. Au lieu de comparer deux individus, on utilise un groupe de I individus (typiquement dix). Si les individus sont soit dominés, soit non dominés, on utilise le partage de la valeur d'efficacité tel qu'il a été décrit dans la section 5.7.2.2.

Le pseudo-code de la fonction sélection pour un problème multiobjectif où tous les objectifs doivent être maximisés est représenté dans l'algorithme 5.7. Son implémentation dans un algorithme génétique est représenté dans l'algorithme 5.8.

Algorithm 5.7 Détail de la fonction de sélection.

```
function selection
 Mélange(random_pop_index)
 Candidat_1 = random_pop_index[1]
 Candidat_2 = random_pop_index[2]
 Candidat_1_dominé = false
 Candidat_2_dominé = false
 for Ensemble_index_comp = 3 to t_dom + 3 do
 /* sélectionne t_dom individus au hasard dans S */
 begin
  Indiv_comp = random_pop_index[Ensemble_index_comp]
  if S[Indiv_comp] domine S[Candidat_1]
    then Candidat_1_dominé = true
  if S[Indiv_comp] domine S[Candidat_2]
    then Candidat_2_dominé = true
 end
if (Candidat_1_dominé AND NOT Candidat_2_dominé)
  then return Candidat_2
  else if (NOT Candidat_1_dominé AND Candidat_2_dominé)
         then return Candidat_1
  else
    do Partage
end
```

La fonction de sélection choisit un individu de la population S.
Les valeurs de t_{dom} et $_{share}$ doivent être fournies par l'utilisateur.

5.7.2.3.3 Discussion

Comme cette méthode n'applique pas la sélection de Pareto sur toute la population, mais

Algorithm 5.8 Algorithme N.P.G.A.

Initialisation de la population
Evaluation des fonctions objectif
For i=1 to G
  Sélection par tournoi entre deux individus
   (utilisation de t_{dom})
    Seul le candidat 1 est dominé : Sélectionner le candidat 2
    Seul le candidat 2 est dominé : Sélectionner le candidat 1
    Les deux candidats sont dominés ou ne sont pas dominés :
      Effectuer un partage d'efficacité
      Sélectionner le candidat avec le plus petit nombre de
       voisins (utilisation de $_{share}$)
  Croisement
  Mutation
  Evaluation des fonctions objectif
End For

seulement sur une partie à chaque exécution, elle est très rapide. Cependant, elle requiert de la part de l'utilisateur les paramètres t_{dom} et $_{share}$, dont dépendent les performances de l'algorithme. Dans [Deb 01], on trouvera des techniques permettant de régler "automatiquement" ces paramètres.

5.7.2.4 La méthode *Weighted Average Ranking G.A.* (W.A.R.G.A)

5.7.2.4.1 Principe

Cette méthode s'inspire de la méthode MOGA. La différence principale réside dans la manière dont la relation de dominance est établie entre deux solutions.

5.7.2.4.2 Présentation de la méthode

Dans cette méthode, le rang de dominance est calculé de la manière décrite dans l'algorithme 5.9, nommée W.A.R. (*Weighted Average Ranking*) :

Algorithm 5.9 La relation W.A.R..

A solution à comparer (vecteur de dimension n).
S ensemble de solutions (S ne contient pas A).

repeat
  $i = 1$
  soit A_i la variable i du point A.
  N_i =nombre de points de S meilleurs que A_i du point de vue de la variable i.
  $i = i + 1$
until $i > n$

Le rang de A vaut $N = \oplus_{i=1}^{n} N_i$

Ensuite, l'efficacité d'un individu sera calculée de la même manière que pour la méthode MOGA.

Prenons un exemple. A la figure 5.17 (f_1 et f_2 sont à minimiser), on a :

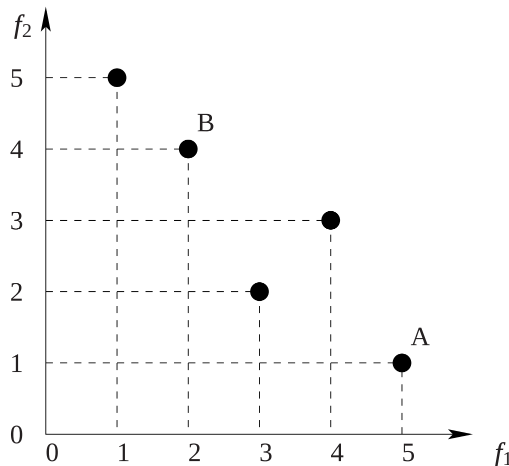


FIG. 5.17 – La relation W.A.R.

- Vis-à-vis de f_1, A est dominé par 4 points.
  Vis-à-vis de f_2, A est dominé par 0 point.
  Donc, $\mathrm{N}(\mathrm{A}) = 4 + 0 = 4$
- Vis-à-vis de f_1, B est dominé par 1 point.
  Vis-à-vis de f_2, B est dominé par 3 points.
  Donc, $N(B) = 1 + 3 = 4$

5.7.2.4.3 Discussion

Pour cette méthode, le nombre de comparaisons à effectuer pour établir les rangs de dominance sur une population de N individus pour une génération est de :

$$N \cdot (N-1) \cdot n \tag{5.21}$$

Cependant, l'algorithme permettant d'implémenter la dominance W.A.R est beaucoup plus simple que celui de la relation de dominance classique.

5.7.2.5 L'élitisme dans les algorithmes génétiques

La méthode SPEA (*Strength Pareto Evolutionary Algorithm*) présentée dans [Zitzler et al. 98] range les solutions non dominées dans une population séparée de la population courante. Ensuite, des individus sont sélectionnés dans la population courante pour une phase de croisement. Les individus avec lesquels ils vont se croiser sont, quant à eux, choisis dans la population des individus non dominés.

Dans [Gandibleux 00], la même méthode est utilisée, à une différence près : la population élitiste est initialisée avec deux individus extrêmes, comme nous l'expliquons ci-dessous :

Le problème de départ est un problème d'optimisation biobjectif.

On part du problème suivant.

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{minimiser} & f_2(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{g}(\overrightarrow{x}) \leq 0 \\ \text{et} & \overrightarrow{h}(\overrightarrow{x}) = 0 \end{array}\right.$$

Le premier individu initial de la population élitiste, soit $\overrightarrow{x_1}$, est obtenu en résolvant :

$$\left| \begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array} \right.$$

Le second individu initial de la population élitiste, soit $\overrightarrow{x_2}$, est obtenu en résolvant :

$$\left| \begin{array}{ll} \text{minimiser} & f_2\left(\overrightarrow{x}\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array} \right.$$

Cette “précaution” permet de “garantir” que la surface de compromis se répartira entre ces deux solutions extrêmes. En effet, les vecteurs objectif $\left(f_1\left(\overrightarrow{x_1}\right), f_2\left(\overrightarrow{x_1}\right)\right)$ et $\left(f_1\left(\overrightarrow{x_2}\right), f_2\left(\overrightarrow{x_2}\right)\right)$ sont non dominés, de génération en génération. Ils transmettront donc leurs caractéristiques aux autres individus. En fin d'optimisation, tous les individus non dominés se répartiront sur toute la surface de compromis.

Dans [Parks et al. 98], un archivage d'individus non dominés est exploité. En début de phase de croisement, les auteurs injectent quelques individus appartenant à l'archive des individus non dominés dans la population normale. Ils concluent que les performances sont meilleures quand la proportion d'individus injectés est plus grande.

Une définition concernant l'élitisme est donnée dans [Laumanns et al. 00].

Définition : élitisme
Soit P^t une population d'un algorithme génétique donné après t itérations (générations). Soit $P^t(x)$ la probabilité qu'a un individu $x \in P^t$ d'être sélectionné pour une phase de croisement et/ou mutation à la génération t. Alors l'algorithme génétique est dit “élitiste” si et seulement si, pour une relation de préférence $\prec$ donnée dans un problème de décision, la condition suivante est vérifiée :
$\forall t \in \mathbb{N}, \exists \overrightarrow{x} \in P^{*t}$ tel que $P^{t+1}\left(\overrightarrow{x}\right) > 0$
avec
$P^{*t} = \left\{ \overrightarrow{x} \in P^t \,|\, \nexists \overrightarrow{x'} \in P^t \,|\, \overrightarrow{x'} \prec \overrightarrow{x} \right\}$
$P^t = \bigcup_{r \leq t} P^r$

L'opération $\bigcup_{r \leq t} P^r$ désigne l'union de tous les ensembles P^r avec $r \leq t$.

P^{*t} désigne donc tous les individus non dominés de P^t.

L'idée générale qui ressort de ces réflexions sur l'élitisme est qu'il est important de ne pas oublier de conserver, d'une manière ou d'une autre, les bonnes solutions obtenues génération après génération.

Pour finir, en optimisation monobjectif, on a pour habitude de sauvegarder la meilleure solution obtenue au cours du processus d'optimisation. En optimisation multiobjectif, un processus similaire consiste à placer dans une archive tous les individus non dominés obtenus en cours d'optimisation. Cette archive représente à tout moment la meilleure approximation de la surface de compromis que l'on a obtenue jusqu'ici. De plus, comme cette archive ne sert qu'à sauvegarder les individus non dominés, elle ne modifie pas le comportement de l'algorithme génétique.

5.8 Bibliographie commentée

[Sait et al. 99] On trouvera dans ce livre la description de certaines métaheuristiques : Recuit simulé, recherche Tabou, algorithmes génétiques, etc. Les méthodes y sont décrites clairement et la plupart des éléments théoriques relatifs à la convergence de certaines de ces méthodes sont présentés. Chaque méthode fait aussi l'objet d'une présentation de son utilisation dans un contexte industriel.

[Goldberg 94] Le livre d'introduction aux algorithmes génétiques de référence. Tout ce qui a trait aux algorithmes génétiques y est présenté de manière didactique.

[Davis 91] Ce livre présente un panorama des applications des algorithmes génétiques.

[Glover et al. 97] Le livre de référence sur la méthode de recherche Tabou. A la différence du livre [Goldberg 94] sur les algorithmes génétiques, cet ouvrage est plus scientifique. L'exposé est un peu plus ardu. Néanmoins, nous conseillons la lecture de ce livre avant toute utilisation de la méthode de recherche Tabou.

[Ausiello et al. 99] Un livre sur la complexité des algorithmes ainsi que sur les méthodes d'approximation. Ce livre traite donc des "performances" des algorithmes. Le niveau de cet ouvrage est assez élevé mais il comporte une présentation très claire des différents concepts mis en œuvre dans ce domaine. Il contient aussi un chapitre "encyclopédique" sur les différents problèmes d'optimisation, ainsi que les principaux résultats s'y rattachant.

[Deb 01] Le livre de référence sur l'utilisation des algorithmes génétiques dans un contexte multiobjectif. Toutes les méthodes utilisant des algorithmes génétiques décrites dans ce chapitre sont aussi présentées dans ce livre. L'auteur propose aussi pour chaque méthode un exemple de fonctionnement "résolu à la main" des différentes méthodes. Ces différents exemples permettent de mieux approfondir la connaissance de ces méthodes.

Chapitre 6

Les méthodes d'aide à la décision

6.1 Introduction

Les méthodes que nous avons présentées jusqu'ici étaient basées sur la relation de dominance. Cette relation (que l'on peut définir de plusieurs manières : la dominance de Pareto, la dominance lexicographique par exemple) permet de filtrer les éléments d'un ensemble, et de ne retenir que les éléments incomparables entre eux. Cependant, il existe une autre approche pour obtenir un ensemble de solutions, qui repose sur l'établissement d'une relation d'ordre entre les différents éléments. Ainsi, on peut, en fonction de la relation d'ordre définie, obtenir un ensemble de solutions (relation d'ordre partiel) ou une et une seule solution (ordre complet). L'autre différence majeure, par rapport aux méthodes d'optimisation multiobjectif "classiques", vient du fait que les méthodes d'aide à la décision ne travaillent que sur des ensembles discrets de points (les méthodes d'optimisation multiobjectif "classiques" peuvent travailler, elles, sur des ensembles continus).

De plus, les méthodes d'aide à la décision permettent de répondre à plusieurs problématiques, réunies dans le tableau 6.1.

Problématique	Résultat	Procédure
Choix d'un sous-ensemble des actions les "meilleures" ou, a défaut, les plus "satisfaisantes".	Choix	Sélection
Tri par affectation des actions à des catégories prédéfinies.	Tri	Affectation
Rangement de classes d'équivalence composées d'actions, ces classes étant ordonnées de façon complète ou partielle.	Rangement	Classement

TAB. 6.1 – Les différentes problématiques.

L'aide à la décision a été développée à partir de la constatation explicitée maintenant.

Dans certains cas, lorsque l'on est amené à comparer trois actions, on peut rencontrer un bouclage entre ces actions. Ce phénomène est appelé l'***intransitivité*** (ce terme sera défini mathématiquement dans le prochain paragraphe) de la préférence, et de l'indifférence, ou ***paradoxe de Condorcet***. Il se résume de la manière suivante : soient trois actions A, B et C,

on peut avoir $A \geq B$, $B \geq C$ et $C \geq A$ (ici, le symbole $\geq$ désigne la préférence). Ce paradoxe est représenté à la figure 6.1.

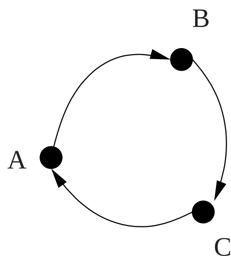


FIG. 6.1 – Le paradoxe de Condorcet.

Un autre exemple reflétant, lui, l'intransitivité de l'indifférence, est celui des sandwichs [Scharlig 85]. On vous propose deux sandwichs, le premier est constitué de pain et de jambon et le second est constitué d'un peu plus de pain et un peu moins de jambon. Ces deux sandwichs sont suffisamment semblables pour que vous n'ayez aucune préférence entre les deux. Vous prenez le sandwich avec un peu plus de pain (car vous préférez avoir un peu plus de pain que de jambon). On vous présente alors, en plus du sandwich que vous avez déjà choisi, un autre sandwich avec encore un peu moins de jambon et un peu plus de pain. Là, vous choisirez encore le sandwich avec un peu plus de pain (on suppose que vous préférez toujours avoir un peu plus de pain que de jambon). Ce processus se répète jusqu'à ce que vous soyez en possession d'un morceau de pain sans jambon.

Durant tout le processus, vous avez toujours eu une préférence vis-à-vis de la solution qui comportait un peu plus de pain. Maintenant, lorsque vous considérez le problème dans son ensemble, il est clair que vous avez une préférence très nette pour le sandwich avec jambon par rapport à la baguette seule.

Donc, comme pour la relation de préférence, la relation d'indifférence est elle aussi intransitive.

C'est cet ensemble de propriétés qui a suscité le développement de l'aide à la décision. En effet, l'optimisation multiobjectif classique ne prenait pas en compte ces propriétés et, dans certains cas, amenait à des solutions incohérentes, lors de la résolution de problèmes d'optimisation "multicritère" (ce terme est souvent préféré au terme "multiobjectif" dans ce contexte de l'aide à la décision).

L'aide à la décision nous propose donc une prise en compte de ces propriétés d'intransitivité des relations d'ordre, et elle nous propose aussi des familles de méthodes destinées à résoudre des problématiques différentes (**S**élection, **T**ri, **R**angement) de celles traitées par l'optimisation multiobjectif classique.

Lorsque l'on est confronté au domaine de l'aide à la décision, on remarque un certain nombre de termes redondants :

– action,
– classement,
– ordre complet ou partiel.

Une "action" désigne un objet, une décision, un candidat ou autre chose encore. C'est sur cette entité que va s'opérer la sélection (ou le tri, ou le classement).

Une méthode d'aide à la décision va donc permettre de choisir la meilleure action, ou de classer des actions en fonction d'un ou plusieurs critères.

En fonction du type de classement que l'on désirera effectuer, on pourra utiliser différents types de règles de classement ou de choix. Ces règles vont définir un ordre complet (si toutes

les actions sont classées) ou partiel (après le classement, il subsiste des actions incomparables, donc non classées).

6.2 Définitions

6.2.1 Les relations d'ordre et d'équivalence

On rappelle, pour commencer, les définitions des relations d'ordre, ainsi que des relations d'équivalence, qui permettront de classer les actions suivant certains critères.

Définition : la relation d'équivalence
Une relation binaire R (relation entre deux éléments) définie sur un ensemble A est une relation d'équivalence si cette relation est :
- **réflexive** : $\forall x \in A, xRx$
- **symétrique** : $\forall\ (x,y) \in A \times A, xRy \Rightarrow yRx$
- **transitive** : $\forall\ (x,y,z) \in A \times A \times A, xRy et yRz \Rightarrow xRz$

L'exemple le plus simple de relation d'équivalence est la relation $=$ sur l'ensemble $\mathbb{N}$ des entiers naturels.

Rappelons maintenant la définition d'une relation d'ordre.

Définition : la relation d'ordre
Une relation binaire R définie sur un ensemble A est une relation d'ordre si cette relation est :
- **réflexive**
- **antisymétrique** : $\forall (x,y) \in A \times A, xRy et yRx \Rightarrow x = y$
- **transitive**

L'exemple le plus simple de relation d'ordre est la relation $\leq$ sur l'ensemble $\mathbb{N}$ des entiers naturels.

Ces différentes définitions vont nous permettre de construire des relations de comparaison (on les appellera aussi des relations de préférence). Ces relations nous permettront, ensuite, d'ordonner les éléments de l'ensemble sur lequel on effectue ces comparaisons.

Il existe plusieurs types d'ordres. Voici leurs définitions :

Définition : préordre
Un préordre est une relation réflexive et transitive.

Définition : ordre
Un ordre est un préordre antisymétrique (c'est donc une relation d'ordre).

Dans un préordre, les ex-aequo sont possibles alors que dans un ordre, il n'y a pas d'ex-aequo possibles.

Voici quelques définitions complémentaires sur le préordre.

Définition : préordre total
Un préordre total est une relation de préordre pour laquelle tous les éléments d'un ensemble peuvent être mis en relation : l'incomparabilité entre deux éléments n'est pas permise.

Un exemple de préordre total est la relation $\leq$ dans l'ensemble $\mathbb{R}$. En effet, soit deux éléments A et B de $\mathbb{R}$, on a soit $A \leq B$, soit $B \leq A$.

Définition : préordre partiel
Un préordre partiel est une relation de préordre pour laquelle certains éléments d'un ensemble ne peuvent pas être mis en relation : l'incomparabilité entre deux éléments est autorisée.

Par exemple, la relation de dominance est une relation de préordre partiel. En effet, si l'on note $A \prec B$ la relation A domine B, il existe des actions pour lesquelles on a ni $A \prec B$ ni $B \prec A$. Ces actions appartiennent à la surface de compromis.

Ces différentes définitions vont nous permettre de "construire" des relations de comparaison entre actions. En fonction de la problématique qui sera posée, on pourra construire soit une relation d'ordre (pour effectuer un classement des actions ou définir un ordre sur cet ensemble de solutions) soit une relation d'équivalence (pour rechercher la meilleure action parmi un ensemble d'actions ou définir un préordre sur cet ensemble de solutions).

6.2.2 Les relations de préférence

Les différentes définitions présentées ci-dessus vont nous permettre de définir des relations de préférence, d'indifférence et d'incomparabilité entre deux actions :

- La relation de **préférence** entre deux actions,
  cette relation, que l'on notera P, signifie une préférence pour une des deux actions. On a aPb si a est préférée à b.
- La relation d'**indifférence** entre deux actions,
  cette relation, que l'on notera I, signifie une indifférence entre les deux actions. On a aIb si il y a indifférence entre a et b.
- La relation d'**incomparabilité** entre deux actions,
  cette relation, que l'on notera R, signifie que deux actions sont incomparables. On a aRb si il y a incomparabilité entre a et b (ce qui signifie que l'on n'a ni aPb, ni aIb, ni bPa).

Ces trois relations $\{P,I,R\}$ définissent ce que l'on appelle une "structure de préférence".

Une structure de préférence peut être complètement caractérisée par la définition d'une relation de "préférence au sens large" :

aSb si aPb ou si aIb.

La relation de préférence au sens large est donc la réunion d'une relation de "préférence au sens strict" (la relation P) et d'une relation d'indifférence (la relation I). La relation de préférence au sens large est aussi appelée "relation de surclassement".

6.2.3 Définition d'un critère

Définition : un critère [Vincke 89]
Un critère est une fonction *g*, définie sur l'ensemble des actions *A*, qui prend ses valeurs dans un ensemble totalement ordonné, et qui représente les préférences de l'utilisateur selon un point de vue.

Par exemple, la moyenne d'un élève est un critère pouvant servir à déterminer la valeur de l'élève.

Selon la méthode, la définition de la "valuation" d'un critère (ou manière de donner une valeur à un critère) peut changer. En revanche, la définition du critère reste identique.

6.2.4 Analyse

Normalement, à la fin de l'application d'une méthode d'aide à la décision, on procède à une phase d'analyse de sensibilité et d'analyse de robustesse. Ces deux analyses permettent de vérifier la "fiabilité" du résultat fourni par la méthode ou la sensibilité du résultat vis-à-vis des paramètres choisis par l'utilisateur.

Définition : analyse de sensibilité [Maystre et al. 94]
"C'est une analyse consistant à répéter l'utilisation de la méthode d'aide à la décision originale en faisant varier les valeurs attribuées à l'origine aux différents paramètres de la méthode, valeurs qui sont souvent empreintes d'un certain arbitraire. Elle vise à définir les paramètres qui conditionnent le plus étroitement la solution choisie, c'est-à-dire où il suffit d'une faible modification pour changer la solution proposée."

Définition : analyse de robustesse [Maystre et al. 94]
"C'est une analyse cherchant à déterminer le domaine de variation de certains paramètres dans lequel un classement ou un choix reste stable. Elle sert à fournir à l'utilisateur une solution robuste, qui l'informe quant à la capacité de la solution à résister à des variations entre la réalité et le modèle censé la représenter."

6.3 Les différentes méthodes

Nous allons présenter différentes méthodes d'aide à la décision.

Nous commencerons par les méthodes ELECTRE (I, IS, II, III, IV et TRI) puis, nous finirons par deux méthodes proches des méthodes ELECTRE : les méthodes PROMETHEE (I et II).

Chacune de ces méthodes traite une problématique bien précise. Les éléments importants qui différencient ces méthodes sont la définition de la relation de classement des actions (chaque méthode exploite une relation de préférence différente) et la méthode d'exploitation des résultats.

6.3.1 Introduction

Avant de commencer la présentation de ces méthodes, nous allons préciser quelques notations, ainsi qu'une méthode de représentation des résultats.

6.3.1.1 Notations

Les méthodes d'aide à la décision permettent de choisir ou de classer des actions en fonction d'un ensemble de critères.

Nous noterons les actions : a_i, $i = 1, \cdots, k$ et les critères : g_j, $j = 1, \cdots, n$.

L'évaluation de l'action a_i suivant le critère g_j sera notée $g_j(a_i)$.

Par exemple, si le critère g_1 est une moyenne de notes de la matière 1, l'expression $g_1(a)$ est la moyenne de l'élève a dans la matière 1. Ce critère prend comme variable un élève et procure la moyenne de l'élève considéré.

Là aussi, chaque méthode d'aide à la décision va fournir sa propre définition de la valeur d'un critère.

6.3.1.2 La représentation

Les méthodes d'aide à la décision utilisent un graphe orienté (ensemble de sommets et d'arcs reliant ces sommets, les arcs étant orientés par des flèches) pour représenter les relations entre les différentes actions.

Dans ce graphe, on commence par introduire les sommets, qui sont les actions considérées, puis on introduit les arcs entre les différents sommets. On trace un arc du sommet a_i vers le sommet a_j si l'action a_i "surclasse" l'action a_j. S'il n'y a aucune relation entre ces deux actions, on ne trace pas d'arc entre elles.

Nous allons traiter un exemple simple. Dans le tableau 6.2 figurent sept actions ainsi que les relations qui les lient. On lit ce tableau dans le sens ligne vers colonne (par exemple, de la ligne a_2 vers la colonne a_1, on a $a_2 S a_1$).

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1							
a_2	S						
a_3	S						
a_4	S	S	S				
a_5	S	S					S
a_6							
a_7							

TAB. 6.2 – Exemples de relations entre actions.

Quand on représente ces différentes relations sur un graphe, on obtient le graphe de la figure 6.2. On remarque que l'action a_6 n'est reliée à aucune autre action. On dit alors qu'elle est incomparable.

De ce graphe, nous allons extraire des informations importantes. Par exemple, nous allons isoler le noyau N du graphe. Une fois ce noyau isolé, on note A/N les actions restantes (qui n'appartiennent pas au noyau).

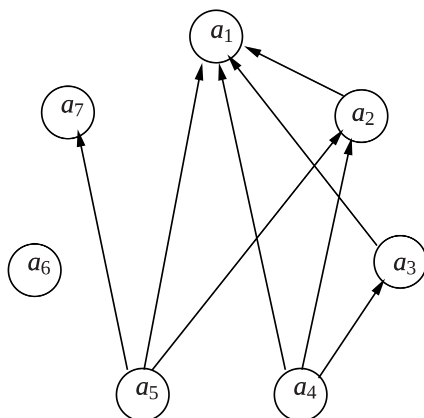
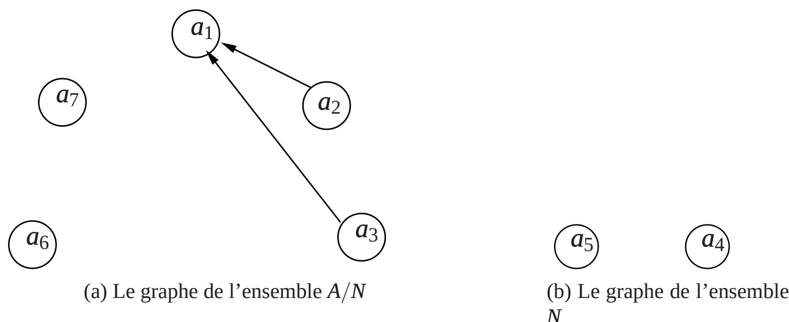


FIG. 6.2 – Graphe des relations entre actions.

Définition : le noyau du graphe [Maystre et al. 94]
Le noyau du graphe est composé d'un ensemble de sommets tels que :
- tous les sommets qui n'appartiennent pas au noyau sont surclassés par un sommet du noyau au moins ;
- les sommets du noyau ne se surclassent pas entre eux.

Si l'on applique la définition à l'exemple, on obtient les graphes de la figure 6.3. Si le surclassement est remplacé par la relation de dominance de Pareto, la définition du noyau correspond alors à la surface de compromis de l'ensemble des solutions.



(a) Le graphe de l'ensemble A/N

(b) Le graphe de l'ensemble N

FIG. 6.3 – Les graphes des ensembles A/N et N.

Cette représentation va permettre, ensuite, de calculer l'indice de concordance et l'indice de discordance, qui permettront de classer ou choisir des actions suivant certains critères.

Définition : indice de concordance [Maystre et al. 94]
"Il est dit du critère j qu'il concorde avec l'hypothèse "l'action a_i surclasse l'action a_k" si l'action a_i est au moins aussi bonne que l'action a_k en ce qui concerne le critère j ; ce qui ce traduit par : $g_j(a_i) \geq g_j(a_k)$."

Définition : non-discordance [Maystre et al. 94]
"La condition de non-discordance permet de refuser le surclassement d'une action sur une autre, obtenu après application de la condition de concordance, lorsqu'il existe une opposition trop forte sur un critère au moins."

Forts de ces différentes définitions, nous allons maintenant pouvoir présenter les méthodes d'aide à la décision.

6.3.2 La méthode ELECTRE I

6.3.2.1 Principe

Cette méthode permet de choisir la meilleure action suivant un groupe de critères. Elle a été définie par B. Roy en 1968 [Vincke 89, Maystre et al. 94].

Cette première méthode sera appliquée à un exemple concret : l'identification du meilleur élève d'une classe.

6.3.2.2 Présentation de la méthode

Notre problème est le suivant :

- On a N actions a_i, $i = 1, \cdots, N$.
- On a K critères g_j, $j = 1, \cdots, K$ (on note $F = \{j \,|\, j = 1, \cdots, K\}$).

On commence par affecter un poids P_j à chaque critère g_j (selon les préférences de l'utilisateur).

Etablir des relations entre actions

On effectue ensuite des comparaisons entre couples d'actions (a_i et a_k). Ces comparaisons seront décomposées en fonction des relations de préférence. On aura les ensembles de comparaison suivants :

- $J^{+}(a_i, a_k) = \left\{ j \in F \,|\, g_j(a_i) > g_j(a_k) \right\}$: c'est l'ensemble des critères pour lesquels l'action a_i est préférée à l'action a_k.
- $J^{=}(a_i, a_k) = \left\{ j \in F \,|\, g_j(a_i) = g_j(a_k) \right\}$
- $J^{-}(a_i, a_k) = \left\{ j \in F \,|\, g_j(a_i) < g_j(a_k) \right\}$

Ces différentes comparaisons sont effectuées sur des couples d'actions (a_i, a_k) et par critère g_j. Le résultat obtenu à la fin de cette étape est un ensemble d'indices représentant des critères qui vérifient les relations définies ci-dessus pour le couple d'actions donné.

Convertir les relations entre actions en valeurs numériques

Le but est, maintenant, de convertir les différents ensembles obtenus suite à la comparaison sur les couples d'actions (a_i, a_k) en valeurs numériques.

Pour chaque ensemble, on détermine donc la somme des poids des critères appartenant à chaque famille :

- $P^{+}(a_i, a_k) = \bigoplus_j P_j$ avec $j \in J^{+}(a_i, a_k)$: c'est la somme des poids des critères appartenant à l'ensemble $J^{+}(a_i, a_k)$.
- $P^{=}(a_i, a_k) = \bigoplus_j P_j$ avec $j \in J^{=}(a_i, a_k)$
- $P^{-}(a_i, a_k) = \bigoplus_j P_j$ avec $j \in J^{-}(a_i, a_k)$

On note $P(a_i, a_k) = P^+(a_i, a_k) + P^=(a_i, a_k) + P^-(a_i, a_k)$.

On pourra représenter toutes ces informations dans un tableau de dimension $N \times N$.

Fusionner les différentes valeurs numériques

Nous allons maintenant passer d'une valeur numérique par couple d'actions et par critère (ce qui fait K valeurs numériques par couple d'actions) à deux valeurs numériques : l'indice de concordance et l'indice de discordance.

Toutes les définitions ci-dessus nous permettent de calculer :

– l'indice de concordance :

$$C_{ik} = \frac{P^+(a_i, a_k) + P^=(a_i, a_k)}{P(a_i, a_k)} \tag{6.1}$$

On a $0 \leq C_{ik} \leq 1$.

Cet indice exprime combien l'hypothèse de départ "a_i surclasse a_k" concorde avec la réalité représentée par les évaluations des actions.

– l'ensemble de concordance :

$$J(a_i, a_k) = J^+(a_i, a_k) \cup J^=(a_i, a_k) \tag{6.2}$$

– l'indice de discordance : on le note D_{ik}. Il est défini de la manière suivante :

$$D_{ik} = \begin{cases} 0 & \text{si } J^-(a_i, a_k) = \emptyset \\ \frac{1}{\Delta_j} \cdot \max(g_j(a_k) - g_j(a_i)) & \text{avec } j \in J^-(a_i, a_k), \text{ sinon} \end{cases} \tag{6.3}$$

et Δ_j : amplitude de l'échelle associée au critère j.

Ici, on a $0 \leq D_{ik} \leq 1$.

– l'ensemble de discordance : c'est l'ensemble $J^-(a_i, a_k)$.

Filtrer les actions

Nous avons maintenant toutes les informations nécessaires pour extraire l'action ou les actions les meilleures.

Pour cela, il va falloir juger les relations de surclassement. Pour qu'une relation de surclassement soit jugée comme fiable, il faut que l'on ait :

$$\left.\begin{array}{l} C_{ik} \geq c \\ D_{ik} \leq d \end{array}\right\} \Leftrightarrow a_i \, S \, a_k \tag{6.4}$$

On a :

c : seuil de concordance. Typiquement $\simeq 0.7$. C'est le seuil au delà duquel l'hypothèse de départ "a_i surclasse a_k" sera considérée comme valable.

d : seuil de discordance. Typiquement $\simeq 0.3$. C'est le seuil en deçà duquel l'hypothèse de départ "a_i surclasse a_k" ne sera plus valable.

Cette définition nous indique que pour qu'une relation de surclassement soit considérée comme valable, il faut que celle-ci soit suffisamment concordante et aussi suffisamment non discordante avec l'hypothèse de départ "a_i surclasse a_k".

Cette dernière étape nous permet d'extraire, de l'ensemble des actions de départ, les meilleures actions (celles qui respectent la relation 6.4).

6.3.2.3 Exemple

Nous allons maintenant illustrer le fonctionnement de la méthode ELECTRE I sur un exemple.

Présentation des données

On considère les notes de cinq élèves dans cinq matières. Ces données sont réunies dans le tableau 6.3.

	M_1	M_2	M_3	M_4	M_5
E_1	7	13	8	12	11
E_2	8	11	11	12	11
E_3	20	2	10	3	15
E_4	16	14	16	14	13
E_5	12	12	8	8	10

TAB. 6.3 – Notes de cinq élèves dans cinq matières.

On cherche à déterminer quel est le meilleur élève suivant trois critères :

C_1 : la moyenne des notes sur l'ensemble des matières doit être la plus élevée possible.

C_2 : la note minimale doit être strictement supérieure à 8.

C_3 : les variations des notes autour de la moyenne doivent être les plus petites possible.

Voici les définitions mathématiques des différents critères :

$$C_1(E) = \frac{\bigoplus_{matière} Note(E, matière)}{5}$$

$$C_2(E) = \begin{cases} 10 & \text{si } \min_{matière}(Note(E, matière)) > 8 \\ 0 & \text{sinon} \end{cases}$$

$$C_3(E) = \frac{\sqrt{\bigoplus_{matière}(Note(E, matière) - C_1(E))^2}}{5}$$

en désignant par :

E : un élève parmi les cinq.

matière : un index qui parcourt toutes les matières pour l'élève considéré (cinq matières en tout).

$Note(E, matière)$: désigne la note de l'élève E dans la matière *matière*.

Les actions seront les suivantes :

a_1 : E_1 est le meilleur élève.

a_2 : E_2 est le meilleur élève.

a_3 : E_3 est le meilleur élève.

a_4 : E_4 est le meilleur élève.

a_5 : E_5 est le meilleur élève.

Il nous faut maintenant choisir des poids pour les différents critères. Par ce choix, on définit un ordre d'importance dans les critères. Par exemple, on veut que la moyenne soit le critère principal et que les deux autres critères aient le même poids.

Donc on fixe $P_{C_1} = 0.5$, $P_{C_2} = P_{C_3} = 0.25$.

Dans un premier temps, on calcule la valeur de chaque critère pour chaque élève. Ces valeurs sont réunies dans le tableau 6.4.

	C_1	C_2	C_3
E_1	10.2	0	1.035
E_2	10.6	10	0.606
E_3	10	0	3.08
E_4	14.6	10	0.5366
E_5	10	10	0.8

TAB. 6.4 – Les valeurs des critères pour chaque élève.

	C'_1	C'_2	C'_3
E_1	0.87	0	16.08
E_2	2.61	20	19.45
E_3	0	0	0
E_4	20	20	20
E_5	0	20	17.9

TAB. 6.5 – Les valeurs recalibrées des critères pour chaque élève.

Les échelles des différents critères ne nous conviennent pas, nous allons donc recalibrer les différentes valeurs :

- Pour le critère C_1, à 10 on associe 0 et à 14.6 on associe 20.
- Pour le critère C_2, à 0 on associe 0 et à 10 on associe 20.
- Pour le critère C_3, à 0.5366 on associe 20 et à 3.08 on associe 0.

Pour les valeurs intermédiaires de chaque critère, on effectue une interpolation linéaire en utilisant les informations que l'on vient de fournir.

On obtient alors les valeurs contenues dans le tableau 6.5.

Ici, les valeurs des amplitudes des échelles (les variables Δ_j) valent toutes 20 (les valeurs de chaque critère sont réparties sur un intervalle $[0,20]$).

Etablir des relations entre actions

Nous allons maintenant effectuer des comparaisons action par action pour chaque critère. Ces comparaisons sont réunies dans un tableau pour chaque critère (voir les tableaux 6.6, 6.7 et 6.8).

La présence du symbole J^- à l'intersection de la ligne a_1 et de la colonne a_2 du tableau relatif au critère C'_1 signifie que l'index 1 appartient à l'ensemble $J^-(a_1,a_2)$. Si l'on réunit les informations des trois tableaux relatifs à ce couple d'actions, on obtient :

$$J^-(a_1,a_2) = \{1,2,3\}$$

En effet, pour les trois critères C'_1, C'_2 et C'_3, l'action a_2 est préférée à l'action a_1.

On considère maintenant toutes les informations contenues dans les tableaux précédents et on calcule les indices de concordance C_{ik}, les indices de discordance D_{ik}, ainsi que les tableaux 6.9 à 6.11 réunissant les informations concernant J^+, $J^=$ et J^-. Ces tableaux réunissent les informations obtenues dans les précédents tableaux, par critère. Par exemple, dans le tableau 6.9, à l'intersection de la colonne a_1 et de la ligne a_2, on trouve l'ensemble $\{1,2,3\}$. Cela signifie que l'action a_2 est préférée à l'action a_1 suivant les critères 1, 2 et 3.

	a_1	a_2	a_3	a_4	a_5
a_1		J^-	J^+	J^-	J^+
a_2	J^+		J^+	J^-	J^+
a_3	J^-	J^-		J^-	$J^=$
a_4	J^+	J^+	J^+		J^+
a_5	J^-	J^-	$J^=$	J^-	

TAB. 6.6 – Comparaisons entre actions pour le critère C'_1.

	a_1	a_2	a_3	a_4	a_5
a_1		J^-	$J^=$	J^-	J^-
a_2	J^+		J^+	$J^=$	$J^=$
a_3	$J^=$	J^-		J^-	J^-
a_4	J^+	$J^=$	J^+		$J^=$
a_5	J^+	$J^=$	J^+	$J^=$	

TAB. 6.7 – Comparaisons entre actions pour le critère C'_2.

	a_1	a_2	a_3	a_4	a_5
a_1		J^-	J^+	J^-	J^-
a_2	J^+		J^+	J^-	J^+
a_3	J^-	J^-		J^-	J^-
a_4	J^+	J^+	J^+		J^+
a_5	J^+	J^-	J^+	J^-	

TAB. 6.8 – Comparaisons entre actions pour le critère C'_3.

	a_1	a_2	a_3	a_4	a_5
a_1		$\emptyset$	$\{1,3\}$	$\emptyset$	$\{1\}$
a_2	$\{1,2,3\}$		$\{1,2,3\}$	$\emptyset$	$\{1,3\}$
a_3	$\emptyset$	$\emptyset$		$\emptyset$	$\emptyset$
a_4	$\{1,2,3\}$	$\{1,3\}$	$\{1,2,3\}$		$\{1,3\}$
a_5	$\{2,3\}$	$\emptyset$	$\{2,3\}$	$\emptyset$	

TAB. 6.9 – Résumé concernant l'ensemble J^+.

	a_1	a_2	a_3	a_4	a_5
a_1		$\emptyset$	$\{2\}$	$\emptyset$	$\emptyset$
a_2	$\emptyset$		$\emptyset$	$\{2\}$	$\{2\}$
a_3	$\{2\}$	$\emptyset$		$\emptyset$	$\{1\}$
a_4	$\emptyset$	$\{2\}$	$\emptyset$		$\{2\}$
a_5	$\emptyset$	$\{2\}$	$\{1\}$	$\{2\}$	

TAB. 6.10 – Résumé concernant l'ensemble $J^=$.

	a_1	a_2	a_3	a_4	a_5
a_1		$\{1,2,3\}$	$\emptyset$	$\{1,2,3\}$	$\{2,3\}$
a_2	$\emptyset$		$\emptyset$	$\{1,3\}$	$\emptyset$
a_3	$\{1,3\}$	$\{1,2,3\}$		$\{1,2,3\}$	$\{2,3\}$
a_4	$\emptyset$	$\emptyset$	$\emptyset$		$\emptyset$
a_5	$\{1\}$	$\{1,3\}$	$\emptyset$	$\{1,3\}$	

TAB. 6.11 – Résumé concernant l'ensemble J^-.

Un indice permet de vérifier la cohérence des informations obtenues : lorsque l'on réunit les trois tableaux en un seul ($J = J^+ \cup J^= \cup J^-$), on doit trouver l'ensemble $\{1,2,3\}$ dans chaque case du tableau.

Convertir les relations entre actions en valeurs numériques

Nous allons maintenant calculer le tableau des ensembles de concordance ($J = J^+ \cup J^=$), le tableau des coefficients de concordance (C_{ik}) et le tableau des coefficients de discordance (D_{ik}).

Le tableau 6.12 contient les ensembles de concordance.

	a_1	a_2	a_3	a_4	a_5
a_1		$\emptyset$	$\{1,2,3\}$	$\emptyset$	$\{1\}$
a_2	$\{1,2,3\}$		$\{1,2,3\}$	$\{2\}$	$\{1,2,3\}$
a_3	$\{2\}$	$\emptyset$		$\emptyset$	$\{1\}$
a_4	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$		$\{1,2,3\}$
a_5	$\{1,2,3\}$	$\{2\}$	$\{1,2,3\}$	$\{2\}$	

TAB. 6.12 – Résumé concernant l'ensemble de concordance $J = J^+ \bigcup J^=$.

Le tableau 6.15 contient les coefficients de concordance. Avant de pouvoir calculer ce tableau, nous allons calculer les coefficients P^+_{ik} et $P^=_{ik}$.

	a_1	a_2	a_3	a_4	a_5
a_1		0	0.75	0	0.5
a_2	1		1	0	0.75
a_3	0	0		0	0
a_4	1	0.75	1		0.75
a_5	0.5	0	0.5	0	

TAB. 6.13 – Résumé concernant les coefficients P^+_{ik}.

	a_1	a_2	a_3	a_4	a_5
a_1		0	0.25	0	0
a_2	0		0	0.25	0.25
a_3	0.25	0		0	0.5
a_4	0	0.25	0		0.25
a_5	0	0.25	0.5	0.25	

TAB. 6.14 – Résumé concernant les coefficients $P^=_{ik}$.

Fusionner les valeurs numériques

Nous allons maintenant calculer le tableau des coefficients de concordance en utilisant la formule suivante :

$$C_{ik} = \frac{P^+_{ik} + P^=_{ik}}{P_{ik}}$$

avec $P_{ik} = 1, \forall i$ et $\forall k$.

Le tableau 6.16 contient les coefficients de discordance. On remarque que Δ_j vaut 20 pour tous les critères.

Filtrer les actions

Nous avons maintenant toutes les informations nécessaires pour réaliser un test de concordance et un test de non discordance. On fixe le seuil c du test de concordance à 0.75. Ce test

	a_1	a_2	a_3	a_4	a_5
a_1		0	1	0	0.5
a_2	1		1	0.25	1
a_3	0.25	0		0	0.5
a_4	1	0.75	1		1
a_5	0.5	0.25	1	0.25	

TAB. 6.15 – Résumé concernant les coefficients de concordance C_{ik}.

	a_1	a_2	a_3	a_4	a_5
a_1		0.087	0	0.196	0.196
a_2	0		0	0.13	0
a_3	0.0435	0.1305		1	0.895
a_4	0	0	0		0
a_5	0.0435	0.0775	0	0.105	

TAB. 6.16 – Résumé concernant les coefficients de discordance D_{ik}.

est satisfait si $C_{ik} \geq 0.75$. On fixe le seuil d du test de non discordance à 0.25. Ce test est satisfait si $D_{ik} \leq 0.25$.

Les C_{ik} qui satisfont le test de concordance sont C_{13}, C_{21}, C_{23}, C_{25}, C_{41}, C_{42}, C_{43}, C_{45} et C_{53}.

Seul D_{34} ne satisfait pas le test de non discordance.

Donc, on peut dire que :

- L'action a_1 surclasse l'action a_3, car la relation de concordance C_{13} est vérifiée et la relation de discordance D_{13} ne l'est pas.
- L'action a_2 surclasse les actions a_1, a_3 et a_5 car les relations de concordance C_{21}, C_{23}, C_{25} sont vérifiées et les relations de discordance D_{21}, D_{23}, D_{25} ne le sont pas.
- L'action a_4 surclasse toutes les autres, car les relations de concordance C_{4i}, $i = 1,2,3,5$ sont vérifiées et les relations de discordance D_{4i}, $i = 1,2,3,5$ ne le sont pas.
- L'action a_5 surclasse l'action a_3, car la relation de concordance C_{53} est vérifiée et la relation de discordance D_{53} ne l'est pas.

On représente ces résultats dans un graphe.

Pour conclure cet exemple, si l'on considère le nombre d'élèves surclassés, on peut dire que l'élève E_4 est meilleur que tout le monde, E_2 se classe second, E_1 et E_5 sont ex-aequo et E_3 est dernier.

6.3.2.4 Discussion

Malgré le nombre important d'étapes que l'on doit franchir pour arriver au résultat, on constate que le choix que l'on peut faire à la fin de l'analyse est très cohérent. Il correspond tout à fait au classement que l'on aurait fait de manière naturelle en appliquant les critères définis.

Cette méthode formalise donc le raisonnement "humain". Ainsi, on peut appliquer cette méthode de choix à un grand nombre de données et obtenir des relations entre actions qu'il aurait été laborieux d'obtenir "à la main".

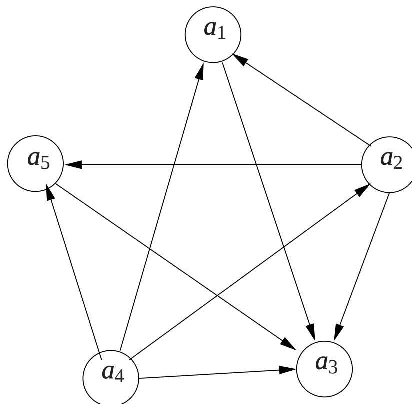


FIG. 6.4 – Graphe des relations entre actions.

6.3.3 La méthode ELECTRE IS

6.3.3.1 Principe

Cette méthode correspond à la méthode ELECTRE I à laquelle on a ajouté l'aspect logique floue [Maystre et al. 94].

6.3.3.2 Présentation de la méthode

Nous allons définir l'indice de concordance par critère, l'indice de concordance global, l'indice de discordance par critère, l'indice de discordance global et la relation de surclassement.

Les indices de concordance et de discordance

– L'indice de concordance :

$$\begin{cases} c_j(a_i, a_k) = 0 & \text{si } p_j < g_j(a_k) - g_j(a_i) \\ 0 < c_j(a_i, a_k) < 1 & \text{si } q_j < g_j(a_k) - g_j(a_i) \leq p_j \\ c_j(a_i, a_k) = 1 & \text{si } g_j(a_k) - g_j(a_i) \leq q_j \end{cases}$$

Cet indice est similaire à celui défini dans la méthode ELECTRE III (voir paragraphe 6.3.5).

– L'indice de concordance global :

$$C_{ik} = \frac{\bigoplus_{j=1}^{m} p_j \cdot c_j(a_i, a_k)}{\bigoplus_{j=1}^{m} p_j}$$

Cet indice est similaire à celui défini dans la méthode ELECTRE III.

– L'indice de discordance par critère ; cet indice est de type binaire :

$$d_j(a_i,a_k) = \begin{cases} 0 & \text{si } g_j(a_k) - g_j(a_i) < v_i(a_i,a_k) - q_j(a_i,a_k) \cdot \frac{1-C_{ik}}{1-c} \\ 1 & \text{sinon} \end{cases}$$

où c est le seuil de concordance global.

– L'indice de discordance global ; cet indice est aussi binaire :

$$D_{ik} = \begin{cases} 0 & \text{si } d_j(a_i,a_k) = 0, \forall j = 1, \cdots, m \\ 1 & \text{sinon} \end{cases}$$

– La relation de surclassement ; elle est aussi binaire :

$$S(a_i,a_k) = \begin{cases} 1 & \text{si } C_{ik} \geq c \text{ et } D_{ik} = 0 \\ 0 & \text{sinon} \end{cases}$$

6.3.3.3 Discussion

Comme avec la méthode ELECTRE I, on peut tracer le graphe des actions et rechercher le noyau.

Il ne faut pas oublier de remplacer le circuit maximal (circuit qui n'est compris dans aucun autre circuit) par une action fictive.

L'analyse des résultats est alors la même qu'avec la méthode ELECTRE I.

L'avantage d'avoir ajouté l'aspect flou à la méthode ELECTRE I est que l'on diminue la sensibilité de la méthode vis-à-vis des paramètres que doit choisir l'utilisateur. En effet, la méthode floue permet de pouvoir choisir un paramètre de la méthode sur une plage, au lieu de donner une valeur ponctuelle.

6.3.4 La méthode ELECTRE II

6.3.4.1 Principe

Cette méthode est très similaire à la méthode ELECTRE I. La différence réside dans la définition de deux relations de surclassement :

– le surclassement fort ;
– le surclassement faible.

6.3.4.2 Présentation de la méthode

Définition des relations de surclassement

Les actions sont comparées deux à deux et des relations de préférence sont établies :

– La relation de surclassement fort : l'action a_i surclasse fortement l'action a_k si :

$$\begin{cases} C_{ik} \geq c^+ \\ \text{et} \\ g_j(a_k) - g_j(a_i) \leq D_1(j), \quad \forall j \\ \text{et} \\ \frac{P_{ik}^+}{P_{ik}^-} \geq 1 \end{cases} \tag{6.5}$$

et/ou

$$\begin{cases} C_{ik} \geq c^0 \\ \text{et} \\ g_j(a_k) - g_j(a_i) \leq D_2(j), \quad \forall j \\ \text{et} \\ \frac{P_{ik}^+}{P_{ik}^-} \geq 1 \end{cases} \tag{6.6}$$

avec $c^+ \geq c^0$ et $D_2(j) \leq D_1(j)$.
Cette relation permet de considérer qu'une action a_i surclasse fortement une action a_k si :
- la relation de surclassement est très fortement concordante et moyennement discordante (équation 6.5),
- la relation de surclassement est moyennement concordante et faiblement discordante (équation 6.6).

– La relation de surclassement faible : l'action a_i surclasse faiblement l'action a_k si :

$$\begin{cases} C_{ik} \geq c^- \\ \text{et} \\ g_j(a_k) - g_j(a_i) \leq D_1(j), \quad \forall j \\ \text{et} \\ \frac{P_{ik}^+}{P_{ik}^-} \geq 1 \end{cases} \tag{6.7}$$

avec $c^0 \geq c^-$.
Cette relation permet de considérer qu'une action a_i surclasse faiblement une action a_k si la relation de surclassement est faiblement concordante et moyennement discordante (équation 6.7).

Les variables $D_1(j)$ et $D_2(j)$ sont des seuils de non discordance. Ces valeurs sont définies pour chaque critère g_j. Ces seuils sont choisis de manière à avoir $D_2(j) \leq D_1(j)$.

En conséquence :
– si $g_j(a_k) - g_j(a_i) \leq D_2(j)$, la discordance est faible et le critère j ne présente pas une opposition majeure à l'hypothèse "l'action a_i surclasse l'action a_k" ;
– si $D_2(j) < g_j(a_k) - g_j(a_i) \leq D_1(j)$, la discordance est moyenne et le critère j ne présente pas une opposition majeure à l'hypothèse de surclassement "l'action a_i surclasse l'action a_k".

Définition du coefficient de concordance

Les coefficients de concordance sont calculés de la même manière que pour la méthode ELECTRE I :

$$C_{ik} = \frac{P_{ik}^+ + P_{ik}^=}{P_{ik}} \tag{6.8}$$

La différence réside dans le fait que l'on aura trois seuils de concordance c^+, c^0 et c^- qui correspondent respectivement à une concordance avec présomption forte, moyenne et faible.

Le test de concordance sera accepté si :

$$\left.\begin{array}{l} C_{ik} \geq c^{+} \\ \text{ou} \\ C_{ik} \geq c^{0} \\ \text{ou} \\ C_{ik} \geq c^{-} \end{array}\right\} \text{ et } \frac{P_{ik}^{+}}{P_{ik}^{-}} \geq 1$$

Le processus de classement

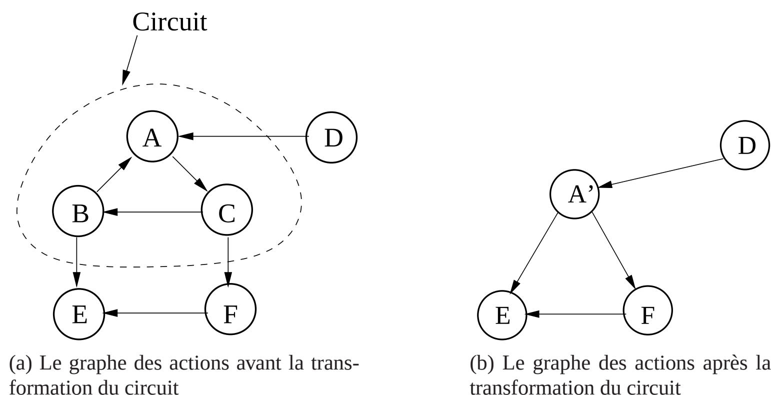
Une fois que l'on a déterminé ces différents éléments, il faut effectuer :
- un classement direct,
- un classement inverse.

On commence par représenter les relations entre actions dans deux graphes :
- un graphe dit de "surclassement fort" noté G_F, dans lequel on lie deux sommets entre eux s'il existe une relation de surclassement fort entre les deux actions représentées par ces deux sommets,
- un graphe dit de "surclassement faible" noté G_f, dans lequel on lie deux sommets entre eux s'il existe une relation de surclassement faible entre les deux actions représentées par ces deux sommets.

Il faut, dans un premier temps, modifier les deux graphes de manière à éliminer les circuits.

Définition : un circuit
Un circuit est un ensemble d'actions dans lequel on a, par exemple, l'action A domine l'action B, l'action B domine l'action C et l'action C domine l'action A.

Un exemple de suppression de circuit est représenté à la figure 6.5. Sur cette figure, on peut voir la simplicité du traitement des circuits. En effet, un circuit correspond à des relations entre les actions correspondantes qui tiennent du paradoxe de Condorcet. Il n'est pas possible de décider quelle action est la meilleure. Donc, pour lever cette ambiguïté, on remplace les actions qui forment un circuit par une action fictive. Cette action symbolisera un groupe d'actions équivalentes pour le décideur.



(a) Le graphe des actions avant la transformation du circuit

(b) Le graphe des actions après la transformation du circuit

FIG. 6.5 – Remplacement d'un circuit par une action fictive.

La modification des graphes s'opère ici comme suit :
- on note G'_F le graphe qui correspond au graphe de surclassement fort d'origine G_F dans lequel on a remplacé tous les circuits par des actions fictives ;

– on note G'_f le graphe qui correspond au graphe de surclassement faible d'origine G_f dans lequel on a remplacé tous les circuits par des actions fictives.

Le classement direct s'effectue en six étapes :

Etape 1 : A chaque nouvelle étape l, les actions déjà classées sont supprimées du graphe de surclassement fort ; les actions restantes constituent l'ensemble A_l (qui est un sous-ensemble de A) et leurs relations sont fournies par le graphe Y_l (qui est un sous-graphe du graphe de surclassement fort modifié G'_F).

Etape 2 : Dans le graphe Y_l, tous les sommets qui ne sont pas surclassés sont recensés : ils forment l'ensemble D.

Etape 3 : Les éléments de D qui sont reliés dans le graphe de surclassement faible G'_f constituent l'ensemble U.

Etape 4 : L'ensemble B contient tous les sommets de U qui ne sont pas surclassés par un autre sommet de U.

La classe d'équivalence des actions classées à la l^e étape, désignée par A_l, est définie par l'union des ensembles $D-U$ et B.

Les ensembles B et $D-U$ représentent tous les sommets qui :

1. n'ont pas encore été classés ;
2. ne sont surclassés par aucun sommet du graphe Y_l, qui n'est autre que le graphe de surclassement fort G'_F duquel ont été enlevés les sommets et arcs correspondants déjà classés ;
3. Concernant $D-U$: n'ont pas de relation de type surclassement faible entre eux.

   Concernant B : surclassent, par des relations de type surclassement faible, d'autres sommets qui remplissent les conditions 1 et 2.

Etape 5 : A toutes les actions classées à l'étape l (et qui forment, par conséquent, la classe d'équivalence A_l) est attribué le rang $l+1$; de cette manière, à chaque action potentielle correspond un rang obtenu par le classement direct et si $rang\,(a_i) < rang\,(a_k)$, cela signifie que l'action a_i est "meilleure" que l'action a_k.

Les sommets classés à la l^e étape sont retirés du graphe de surclassement fort, ce qui donne le nouveau sous-graphe Y_{l+1}.

Etape 6 : Enfin, si Y_{l+1} ne comporte plus aucun sommet, le classement est terminé ; dans le cas contraire, il continue avec l'étape $l+1$.

Etablir la relation d'ordre

Une fois le classement direct effectué, on construit un autre classement : le classement inverse. Pour cela, on effectue un classement direct mais en y apportant les modifications suivantes :

– inverser la direction des arcs dans les graphes G'_F (graphe de surclassement fort) et G'_f (graphe de surclassement faible) ;
– une fois le rang obtenu de la même manière que dans la procédure précédente ($rang_1\,(a_i) = l+1$), on l'ajuste de la manière suivante :

$$rang_2\,(a_i) = 1 + rang_1\,(a_i)_{max} - rang_1\,(a_i)$$

Nous allons maintenant réunir les informations obtenues dans le classement direct et dans le classement inverse (il s'agit, en fait, d'une intersection entre les deux classements). Nous obtiendrons alors un préordre partiel.

Il faut suivre les étapes suivantes :

– si a_i est préférée à a_k dans les deux préordres complets (le classement direct et le classement inverse), alors il en sera de même pour le préordre partiel ;
– si a_i est équivalente à a_k dans un préordre complet, mais si elle lui est préférée dans le deuxième, alors a_i sera préférée à a_k dans le classement final ;
– si, dans le premier préordre, a_i est préférée à a_k et si, dans le deuxième préordre, a_k est préférée à a_i, alors les deux actions seront incomparables dans le préordre final.

6.3.4.3 Exemple

On reprend l'exemple du paragraphe 6.3.2.3.

On détermine les coefficients $\frac{P_{ik}^+}{P_{ik}^-}$ (voir tableau 6.17).

	a_1	a_2	a_3	a_4	a_5
a_1		0	$+\infty$	0	1
a_2	$+\infty$		$+\infty$	0	$+\infty$
a_3	0	0		0	0
a_4	$+\infty$	$+\infty$	$+\infty$		$+\infty$
a_5	1	0	$+\infty$	0	

TAB. 6.17 – Valeurs des coefficients $\frac{P_{ik}^+}{P_{ik}^-}$.

Nous calculons maintenant les coefficients :
– $g_1(a_k) - g_1(a_i)$ que l'on note G_{ik}^1 (voir tableau 6.18).
– $g_2(a_k) - g_2(a_i)$ que l'on note G_{ik}^2 (voir tableau 6.19).
– $g_3(a_k) - g_3(a_i)$ que l'on note G_{ik}^3 (voir tableau 6.20).

	a_1	a_2	a_3	a_4	a_5
a_1		1.74	−0.87	19.13	−0.87
a_2	−1.74		−2.61	17.39	−2.61
a_3	0.87	2.61		20	0
a_4	−19.13	−17.39	−20		−20
a_5	0.87	2.61	0	20	

TAB. 6.18 – Valeurs des coefficients G_{ik}^1.

	a_1	a_2	a_3	a_4	a_5
a_1		20	0	20	20
a_2	−20		−20	0	0
a_3	0	20		20	20
a_4	−20	0	−20		0
a_5	−20	0	−20	0	

TAB. 6.19 – Valeurs des coefficients G_{ik}^2.

On calcule la matrice des relations de surclassement, en utilisant les seuils suivants : $c^+ = 0.75$, $c^0 = 0.7$, $c^- = 0.65$, $D_1 = 20$ et $D_2 = 16$ (voir tableau 6.21). Dans ce tableau, le symbole S_F désigne la relation de surclassement fort, le symbole S_f désigne la relation

	a_1	a_2	a_3	a_4	a_5
a_1		3.37	−16.08	3.92	1.82
a_2	−3.37		−19.45	0.55	−1.55
a_3	16.08	19.45		20	17.9
a_4	−3.92	−0.55	−20		−2.1
a_5	−1.82	1.55	−17.9	2.1	

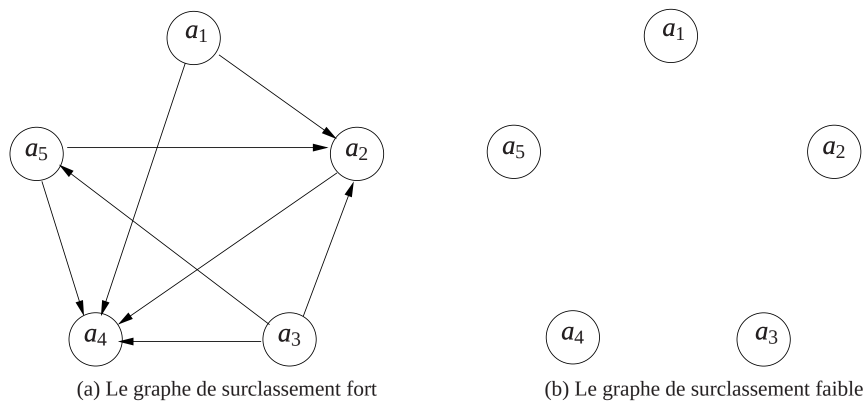
TAB. 6.20 – Valeurs des coefficients G^3_{ik}.

de surclassement faible et $\times$ désigne l'absence de relation de surclassement entre les deux éléments.

	a_1	a_2	a_3	a_4	a_5
a_1		$\times$	$\times$	$\times$	$\times$
a_2	S_F		S_F	$\times$	S_F
a_3	$\times$	$\times$		$\times$	$\times$
a_4	S_F	S_F	S_F		S_F
a_5	$\times$	$\times$	S_F	$\times$	

TAB. 6.21 – Matrice des relations de surclassement.

On trace maintenant le graphe de surclassement fort et le graphe de surclassement faible. Ces graphes sont représentés à la figure 6.6.



(a) Le graphe de surclassement fort (b) Le graphe de surclassement faible

FIG. 6.6 – Les graphes de surclassement.

On effectue maintenant le classement :
- direct (voir tableau 6.22) ;
- inverse (voir tableau 6.23).

Pour finir, on représente les informations obtenues par les classements direct et inverse sur un graphique (voir figure 6.7). En abscisse, on a le rang obtenu lors du classement inverse et, en ordonnée, on a le rang obtenu lors du classement direct.

Étape	Y_l	D	U	B	A_l	r_{l+1}
0	$\{1,2,3,4,5\}$	$\{4\}$	$\emptyset$	$\{4\}$	$\{4\}$	1
1	$\{1,2,3,5\}$	$\{2\}$	$\emptyset$	$\{2\}$	$\{2\}$	2
2	$\{1,3,5\}$	$\{1,2\}$	$\emptyset$	$\{1,5\}$	$\{1,5\}$	3
3	$\{3\}$	$\{3\}$	$\emptyset$	$\{3\}$	$\{3\}$	4

TAB. 6.22 – Le classement direct.

Étape	Y_l	D	U	B	A_l	r_{l+1}
0	$\{1,2,3,4,5\}$	$\{1,3\}$	$\emptyset$	$\{1,3\}$	$\{1,3\}$	4
1	$\{2,4,5\}$	$\{5\}$	$\emptyset$	$\{5\}$	$\{5\}$	3
2	$\{2,4\}$	$\{4\}$	$\emptyset$	$\{4\}$	$\{4\}$	2
3	$\{4\}$	$\{2\}$	$\emptyset$	$\{2\}$	$\{2\}$	1

TAB. 6.23 – Le classement inverse.

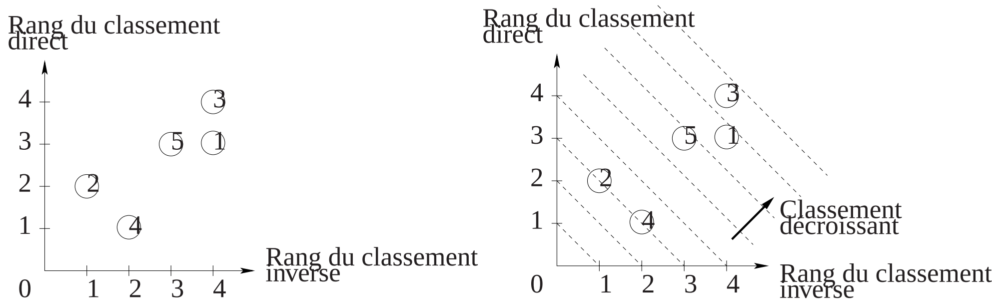


FIG. 6.7 – Classement des actions.

6.3.4.4 Discussion

Cette méthode permet de classer les actions. Dans notre exemple, on a effectué deux classements :

- un classement direct (ou de la meilleure action à la moins bonne) :
  2, 4, 5, $\{3,1\}$.
- un classement inverse (ou de la moins bonne action à la meilleure) :
  4, 2, $\{5,1\}$, 3.
  Nous présentons les actions de la meilleure à la moins bonne pour que les différences de classement puissent être notées.

Le classement global, tel qu'il est donné à la figure 6.7, donne :

$\{2,4\},5,1,3.$

Ce classement a été obtenu en faisant glisser la droite

$$\text{Rang du classement direct } = -\text{Rang du classement inverse}$$

de la droite vers la gauche. Les meilleures actions sont celles que l'on rencontre en premier.

Cette méthode répond bien à la problématique du rangement.

Cependant, cette méthode est complexe à appliquer (l'exemple le montre bien). La méthode travaille avec un nombre d'ensembles important (Y_l, D, B, U et A_l). De plus, elle pos-

sède de nombreux paramètres (les seuils de concordance c^+, c^0 et c^- et les seuils de discordance par critère $D_1(j)$ et $D_2(j)$).

De plus amples informations sur cette méthode peuvent être trouvées dans [Vincke 89] et [Maystre et al. 94].

6.3.5 La méthode ELECTRE III

6.3.5.1 Principe

Cette méthode reprend les mêmes principes que la méthode ELECTRE II (voir [Vincke 89] et [Maystre et al. 94]). La grande différence réside dans l'introduction de pseudo-critères à la place des critères classiques. Pour ces critères, deux seuils seront définis :

– un seuil d'indifférence,
– un seuil de préférence stricte.

ELECTRE III introduit aussi un degré de crédibilité (ou degré de fiabilité) du surclassement. La décision finale n'est plus un choix binaire entre l'acceptation ou le rejet d'une action.

6.3.5.2 Présentation de la méthode

Cette méthode ELECTRE est finalement fortement inspirée de la logique floue.

Définissons ce qu'est un pseudo-critère.

Définition : le pseudo-critère
Un pseudo-critère est défini par deux bandes qui correspondent à deux zones de préférence différentes :
– une bande d'indifférence, dans laquelle la différence entre a_i et a_k n'entraîne pas de préférence particulière pour l'une ou l'autre des deux actions ;
– une bande de préférence stricte, dans laquelle on peut dire que l'on préfère l'une ou l'autre des deux actions.

Entre ces deux bandes, on peut dire que l'on a une préférence faible pour une action.

Le pseudo-critère est donc une fonction g dont le pouvoir discriminant est caractérisé par deux seuils $q(g)$ et $p(g)$, de la façon suivante :

$\forall a_i, a_k \in A$

La relation d'indifférence est définie par :

$a_i \, I \, a_k \Leftrightarrow -q(g(a_i)) \leq g(a_i) - g(a_k) \leq q(g(a_k))$

La relation de préférence faible est définie par :

$a_i \, Q \, a_k \Leftrightarrow q(g(a_k)) < g(a_i) - g(a_k) \leq p(g(a_k))$

La relation de préférence stricte est définie par :

$a_i \, P \, a_k \Leftrightarrow p(g(a_k)) < g(a_i) - g(a_k)$

Les seuils p et q doivent respecter les trois relations suivantes :

$$\frac{q(g(a_k) - g(a_i))}{g(a_k) - g(a_i)} \geq -1$$

$$\frac{p(g(a_k) - g(a_i))}{g(a_k) - g(a_i)} \geq -1$$

$$q_j \leq p_j$$

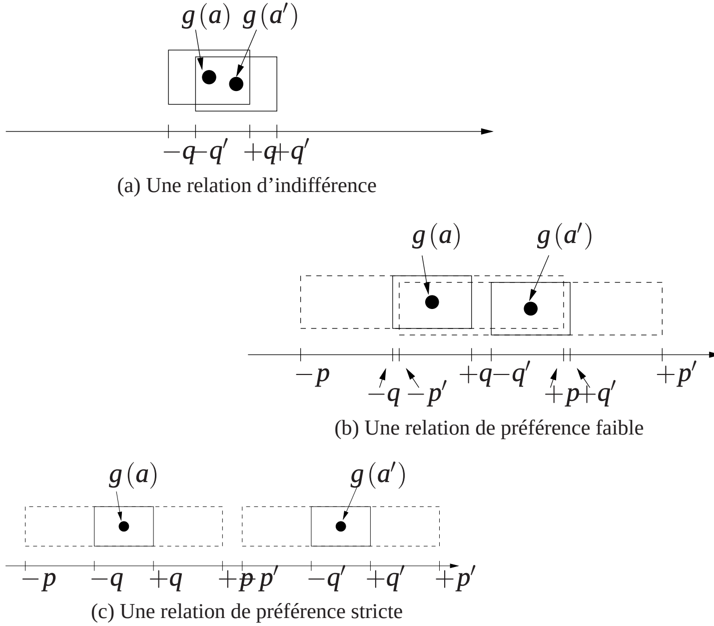
Le paramètre p_j (la fonction seuil p associée au critère j) est appelé le "seuil de préférence stricte" et le paramètre q_j (la fonction seuil q associée au critère j) est appelé le "seuil d'indifférence".

Par exemple, les fonctions seuil p et q peuvent être définies comme :

– une constante,
– une fonction de l'action considérée, par exemple :

$$p\left(g\left(a_i\right)\right) = \quad + \quad \cdot g\left(a_i\right) \tag{6.9}$$

Le fonctionnement de ces différentes relations (I, Q, et P) est représenté à la figure 6.8.



(a) Une relation d'indifférence

(b) Une relation de préférence faible

(c) Une relation de préférence stricte

FIG. 6.8 – Illustration graphique des différentes relations de préférence.

L'indice de concordance

Pour l'indice de concordance, on utilise la formule suivante, illustrée par la figure 6.9 :

$$\begin{cases} c_j\left(a_i, a_k\right) = \frac{g_j(a_i)+p_j-g_j(a_k)}{p_j-q_j} & \text{si } q_j < g_j\left(a_k\right) - g_j\left(a_i\right) \leq p_j \\ c_j\left(a_i, a_k\right) = 1 & \text{si } g_j\left(a_k\right) - g_j\left(a_i\right) \leq q_j \\ c_j\left(a_i, a_k\right) = 0 & \text{si } p_j < g_j\left(a_k\right) - g_j\left(a_i\right) \end{cases}$$

L'indice de concordance global est défini par :

$$C_{ik} = \frac{\bigoplus\limits_j p_j \cdot c_j\left(a_i, a_k\right)}{\bigoplus\limits_j p_j} \tag{6.10}$$

Sur l'axe des abscisses de la figure 6.9, nous avons représenté la différence entre deux actions pour un critère donné $g_1(a_1) - g_1(a_2)$.

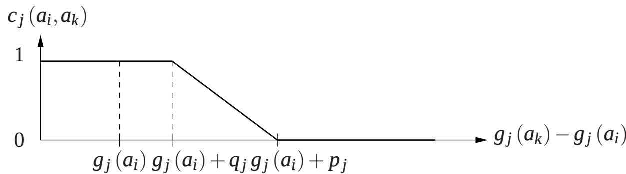


FIG. 6.9 – Représentation de l'indice de concordance.

L'indice de discordance

On définit le seuil de veto v_j. C'est la valeur de $g_j(a_k) - g_j(a_i)$ pour laquelle il apparaît prudent de refuser toute crédibilité de surclassement de l'action a_k par rapport à l'action a_i.

L'indice de discordance est défini par la formule suivante, illustrée par la figure 6.10 :

$$\begin{cases} d_j(a_i, a_k) = 1 & \text{si } v_j < g_j(a_k) - g_j(a_i) \\ d_j(a_i, a_k) = \frac{g_j(a_k) - g_j(a_i) - p_j}{v_j - p_j} & \text{si } p_j \leq g_j(a_k) - g_j(a_i) \leq v_j \\ d_j(a_i, a_k) = 0 & \text{si } g_j(a_k) - g_j(a_i) < p_j \end{cases}$$

On a $q_j < p_j < v_j$.

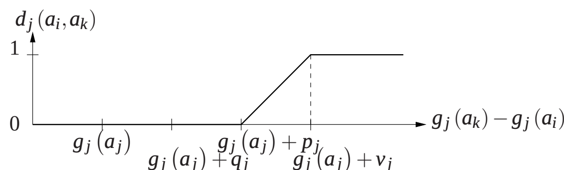


FIG. 6.10 – Représentation de l'indice de discordance.

Définition : degré de crédibilité du surclassement
On définit le degré de crédibilité du surclassement par :

$$\Delta_{ik} = C_{ik} \cdot \prod_{j \in \bar{F}} \frac{1 - d_j(a_i, a_k)}{1 - C_{ik}} \tag{6.11}$$

Dans [Roy et al. 93], on trouvera les règles que doit respecter la fonction Δ_{ik} donnant le degré de crédibilité du surclassement.

Si F est l'ensemble des critères, alors on a :

$$\bar{F} = \{ j \mid j \in F, d_j(a_i, a_k) > C_{ik} \} \tag{6.12}$$

C'est l'ensemble des critères pour lesquels l'indice de discordance est supérieur à l'indice de concordance global.

Exploitation

Pour exploiter la relation de surclassement flou définie plus loin, on définit le seuil de discrimination. C'est un seuil qui permettra de distinguer les relations de surclassement à prendre en compte lors du classement [Maystre et al. 94].

– Seuil de discrimination : $S(\ )$ est une fonction définie pour toute valeur $\ \in [0,1]$.
  On calcule ce seuil en considérant les couples d'actions (a_e, a_m) et (a_i, a_j) :
  1. Calcul de :
     $$\begin{cases} \ = \Delta_{ij} \\ \ - \ = \Delta_{em} \end{cases}$$
     $= \Delta_{ij} - \Delta_{em}$
  2. Appliquer la discrimination :

  si $\ > S(\ )$, alors le surclassement de a_j par a_i est strictement plus crédible que le surclassement de a_m par a_e.
– Exemple de calcul de la fonction seuil :
  Nous présentons ici un exemple tiré de [Roy et al. 93].
  Dans un premier temps, nous allons comparer des valeurs importantes de crédibilité de surclassement.
  On considère que le surclassement de crédibilité 0.99 est plus crédible que le surclassement de crédibilité 0.80. En revanche, on considère que le surclassement de crédibilité 0.99 n'est pas forcément plus crédible que le surclassement de crédibilité 0.85.
  Nous allons maintenant comparer deux classements de faible degré de crédibilité.
  On considère que le surclassement de crédibilité 0.30 est plus crédible que le surclassement de crédibilité 0. En revanche, on considère que le surclassement de crédibilité 0.25 n'est pas plus crédible que le surclassement de crédibilité 0.
  Ceci se traduit par l'ensemble d'équations suivant :
  $0.99 - 0.80 > S(0.99 - 0.80)$
  $0.99 - 0.85 < S(0.99 - 0.85)$
  $0.25 - 0 < S(0.25 - 0)$
  $0.30 - 0 > S(0.30 - 0)$
  On fait maintenant l'hypothèse que $S(\ )$ a la forme suivante : $S(\ ) = A - B \cdot \ $.
  Les paramètres $A = 0.3$ et $B = 0.15$ permettent de vérifier toutes les inéquations précédentes.
  S'il y avait eu beaucoup plus d'inéquations, il aurait peut-être fallu choisir une autre forme pour $S(\ )$; une forme du second degré par exemple. Il faut seulement que la fonction $S(\ )$ soit monotone et non décroissante.
– Exemple d'application :
  Nous allons maintenant appliquer une autre fonction de seuil à la vérification de la crédibilité d'un surclassement par rapport à un autre.
  $S(\ ) = 0.2 - 0.15 \cdot$
  Soit deux groupes d'actions : (a_1, a_2) et (a_2, a_1). Les indices de crédibilité du surclassement : $\Delta_{12} = 0.12$ et $\Delta_{21} = 0.33$.
  $\ = 0.12 - 0.33 = -0.21$
  $S(\ ) = S(0.12) = 0.2 - 0.15 \cdot 0.12 = 0.182$
  $\ < S(\ )$ donc le surclassement de a_2 par a_1 n'est pas plus crédible que le surclassement de a_1 par a_2.

Terminologie liée à la méthode ELECTRE III

Avant de présenter la méthode de classement, on introduit quelques définitions :

Définition : puissance d'une action
c'est le nombre des actions auxquelles elle est strictement préférée. On note $p(a_i)$ la puissance de l'action a_i.

Définition : faiblesse d'une action
c'est le nombre des actions qui lui sont strictement préférées. On note $f(a_i)$ la faiblesse de l'action a_i.

Définition : qualification d'une action
on note $q(a_i) = p(a_i) - f(a_i)$.

Définition : la relation de surclassement flou

$$a_i S_A^{1} a_k \Leftrightarrow \begin{cases} \Delta_{ik} > \;_1 & et \\ \Delta_{ik} > \Delta_{ki} + S(\Delta_{ik}) \end{cases} \quad (6.13)$$

Définition : la $_l$-puissance

$$p_A^{l}(a_i) = card\left(\left\{a_k \in A \,|\, a_i S_A^{l} a_k\right\}\right) \quad (6.14)$$

Définition : la $_l$-faiblesse

$$f_A^{l}(a_i) = card\left(\left\{a_k \in A \,|\, a_k S_A^{l} a_i\right\}\right) \quad (6.15)$$

Définition : la $_l$-qualification de l'action a_i rapportée à l'ensemble A

$$q_A^{l}(a_i) = p_A^{l}(a_i) - f_A^{l}(a_i) \quad (6.16)$$

$card(A)$ désigne le cardinal de l'ensemble A.

Le processus de classement

Voici maintenant la description de la méthode de classement. On appelle cette méthode "chaîne de distillation descendante", car c'est un processus itératif qui a pour effet d'extraire les solutions de -qualification maximale.

Etape 1 : on pose $A_n = A$ et $n = 0$.

Etape 2 : on pose $D_l = A_n$ et $l = 0$.

Etape 3 : $_{l} = \max \Delta_{ik}$ où $a_i, a_k \in A$ et $a_i \neq a_k$.
$S(\ _{l}) = \ + \ \cdot \ _{l}$.

Etape 4 : $_{l+1} = \max \Delta_{ik}$ où $\Delta_{ik} < \ _{l} - S(\ _{l})$ et $a_i, a_k \in D_l$.

Etape 5 : on applique la relation de surclassement flou $a_i S_A^{l+1} a_k$.
On calcule la $_{l+1}$-puissance, la $_{l+1}$-faiblesse et la $_{l+1}$-qualification pour chaque action $a_i \in D_l$.
On calcule $\overline{q} = \max q^{\ l+1}(a_i), a_i \in D_l$.
On calcule $D_{l+1} = \left\{a_i \in D_l \,|\, q^{\ l+1}(a_i) = \overline{q}\right\}$.

Etape 6 : $card\,(D_{l+1}) = 1$ où $_{l+1} = 0$?
Si oui, on incrémente l de 1 et on va à l'étape 4.
Si non, on passe à l'étape 7.

Étape 7 : $C_{n+1} = D_{l+1}$ et $A_{n+1} = A_n / C_{n+1}$ (A_{n+1} est égal à l'ensemble A_n privé des éléments de C_{n+1}).

Etape 8 : $A_{n+1} = \emptyset$?
Si oui, alors le classement est terminé.
Si non, on incrémente n de 1 et on va à l'étape 2.

Pour la méthode de distillation ascendante, on reprend le même cheminement, en changeant les éléments suivants :

- à l'étape 5, on remplace "on calcule $\overline{q} = \max q^{\ l+1}(a_i), a_i \in D_l$" par "on calcule $\underline{q} = \min q^{\ l+1}(a_i), a_i \in D_l$ " ;
- à l'étape 5, on remplace "on calcule $D_{l+1} = \left\{a_i \in D_l \,|\, q^{\ l+1}(a_i) = \overline{q}\right\}$." par "on calcule $D_{l+1} = \left\{a_i \in D_l \,|\, q^{\ l+1}(a_i) = \underline{q}\right\}$.".

Représentation des résultats

A la fin des processus de distillations ascendante et descendante, on obtient deux listes d'actions, associées à des rangs correspondant à l'itération au cours de laquelle l'action considérée a été extraite. Pour avoir une idée des relations qui existent entre ces différentes actions, il suffit de représenter ces actions dans un graphe donnant le rang de distillation ascendante en fonction du rang de distillation descendante.

Par exemple, si après les processus de distillation on obtient la liste des actions représentée dans le tableau 6.24, la représentation de ces actions donnera le graphe de la figure 6.11.

Action	Rang issu de la distillation ascendante	Rang issu de la distillation descendante
a_1	1	3
a_2	2	2
a_3	4	4
a_4	3	1

TAB. 6.24 – Résultat des processus de distillation.

6.3.5.3 Discussion

Cette méthode est très paramétrable. C'est aussi un de ses principaux inconvénients. Il est difficile, même pour un expert, de trouver de bons coefficients et de justifier leurs choix.

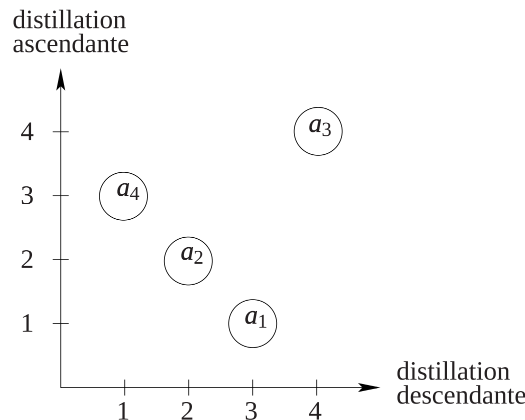


FIG. 6.11 – Représentation des actions après les processus de distillation.

Cependant, ELECTRE III offre à l'utilisateur un outil de représentation 2D des solutions dont la dimension est, en général, supérieure à 2 (la dimension d'une action est égale au nombre de critères).

Comme on pourra le remarquer, les actions qui ont une faible valeur de qualification (rappelons que la qualification est la différence entre le nombre de critères qui "votent" en faveur de notre action et le nombre de critères qui votent en défaveur de notre action) vont obtenir un classement après distillation directe (elles seront plutôt en queue de classement) qui sera complètement différent du classement obtenu après distillation inverse (elles seront en tête de classement). On qualifie ces actions d'actions *flottantes*. Ce sont des actions sur lesquelles il est difficile de faire des choix. En effet, comme la qualification de ces actions est faible, elles ont presque autant de critères qui votent en leur faveur que de critères qui votent en leur défaveur.

6.3.6 La méthode ELECTRE IV

6.3.6.1 Principe

C'est aussi une méthode destinée au classement d'actions. Elle est directement inspirée de la méthode ELECTRE III. La grande différence vient du fait que l'on n'a plus à donner de poids pour chaque critère [Vincke 89, Maystre et al. 94].

6.3.6.2 Présentation de la méthode

On commence par comparer les actions deux à deux.

On définit les notations suivantes :

- $m_p(a_i, a_k)$ est le nombre de critères pour lesquels l'action a_i est strictement préférée à l'action a_k.
- $m_q(a_i, a_k)$ est le nombre de critères pour lesquels l'action a_i est strictement préférée à l'action a_k.
- $m_{in}(a_i, a_k)$ est le nombre de critères pour lesquels les actions a_i et a_k sont considérées comme indifférentes, bien que a_i ait une meilleure évaluation que a_k.
- $m_o(a_i, a_k) = m_o(a_k, a_i)$ est le nombre de critères pour lesquels les actions a_i et a_k ont la même évaluation.

Si m est le nombre total de critères, on doit avoir :

$$m = m_p(a_i,a_k) + m_q(a_i,a_k) + m_{in}(a_i,a_k) + m_o(a_i,a_k) + m_{in}(a_k,a_i) + m_q(a_k,a_i) + m_p(a_k,a_i)$$

On peut remarquer une certaine correspondance entre ces notations et les notations introduites dans la méthode ELECTRE III. Cette correspondance est illustrée par la figure 6.12.

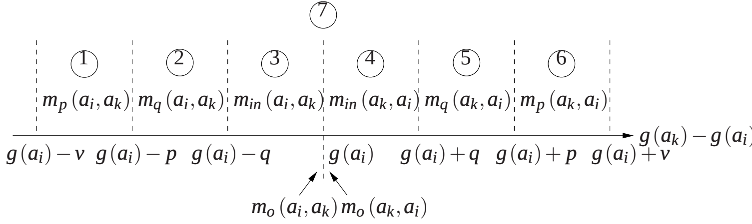


FIG. 6.12 – Correspondance entre les notations des méthodes ELECTRE III et ELECTRE IV.

Zone 1 : l'action a_i est strictement préférée à l'action a_k ($a_i P a_k$).

Zone 2 : l'action a_i est faiblement préférée à l'action a_k ($a_i Q a_k$).

Zone 3 : l'action a_i est à peine préférée à l'action a_k ($a_i I a_k$).

Zone 4 : l'action a_k est à peine préférée à l'action a_i ($a_k I a_i$).

Zone 5 : l'action a_k est faiblement préférée à l'action a_i ($a_k Q a_i$).

Zone 6 : l'action a_k est strictement préférée à l'action a_i ($a_k P a_i$).

Zone 7 : les actions a_i et a_k sont indifférentes.

Définitions liées à la méthode ELECTRE IV

On définit ensuite quatre niveaux de dominance dans la relation de surclassement :

Définition : quasi-dominance

On note cette relation S_q : a_i surclasse a_k avec quasi-dominance si

$$a_i S_q a_k \Leftrightarrow \begin{cases} m_p(a_k,a_i) + m_q(a_k,a_i) = 0 & \text{et} \\ m_{in}(a_k,a_i) \leq 1 + m_{in}(a_i,a_k) + m_p(a_i,a_k) + m_q(a_i,a_k) & \end{cases}$$

Définition : dominance canonique

On note cette relation S_c : a_i surclasse a_k avec dominance canonique si

$$a_i S_c a_k \Leftrightarrow \begin{cases} m_p(a_k,a_i) = 0 & \text{et} \\ m_q(a_k,a_i) \leq m_p(a_i,a_k) & \text{et} \\ m_q(a_k,a_i) + m_{in}(a_i,a_k) \leq 1 + m_{in}(a_i,a_k) + m_q(a_i,a_k) + m_p(a_i,a_k) & \end{cases}$$

Définition : pseudo-dominance
On note cette relation S_p : a_i surclasse a_k avec pseudo-dominance si

$$a_i \, S_p \, a_k \Leftrightarrow \begin{cases} m_p\left(a_k, a_i\right) = 0 & \text{et} \\ m_q\left(a_k, a_i\right) \leq m_q\left(a_i, a_k\right) + m_p\left(a_i, a_k\right) & \end{cases}$$

Définition : veto-dominance
On note cette relation S_v : a_i surclasse a_k avec veto-dominance si

$$a_i \, S_v \, a_k \Leftrightarrow \begin{cases} m_p\left(a_k, a_i\right) = 0 & \text{ou} \\ m_q\left(a_k, a_i\right) = 1 & \text{et} \\ \text{non } a_k \, P_{v_j} \, a_i, \, \forall j & \text{et} \\ m_p\left(a_i, a_k\right) \geq \frac{m}{2} & \end{cases}$$

Exploitation de la méthode

Ensuite, une fois les relations de surclassement déterminées, on passe aux relations de surclassement flou, en associant un degré de crédibilité à S_q, S_c, S_p et S_v.

Pour finir, on applique la même méthode de classement (distillation ascendante et distillation descendante) que pour la méthode ELECTRE III.

6.3.6.3 Discussion

Cette méthode a pour principal avantage de faire disparaître les poids associés aux différents critères. En fin d'analyse, on retrouve, mais de manière moins forte, l'inconvénient des poids des critères, par l'intermédiaire du choix du degré de crédibilité associé à chaque relation S_q, S_c, S_p et S_v.

L'avantage réside dans le fait qu'il n'y a que quatre coefficients à choisir, alors que l'on peut avoir beaucoup plus de critères (et donc beaucoup plus de poids à déterminer dans la méthode ELECTRE III). Néanmoins, on retrouve la complexité de la méthode ELECTRE III.

6.3.7 La méthode ELECTRE TRI

6.3.7.1 Principe

Cette méthode permet de classer des actions dans différentes catégories [Maystre et al. 94]. Les catégories sont séparées par des “actions de référence”. Comme la comparaison se fait entre les actions et les actions de référence, cette méthode permet de traiter un plus grand nombre d'actions. Par exemple, si l'on a 100 actions et 5 actions de référence, avec la méthode ELECTRE TRI, on devra faire 100×5 comparaisons. Avec une autre méthode, on devra faire 100×99 comparaisons (chaque action étant comparée à toutes les autres).

6.3.7.2 Présentation de la méthode

Le choix des actions de référence

On choisit des actions parfaitement comparables entre elles : chacune des actions surclasse ou est surclassée par toutes les autres. On parle alors de segmentation multiobjectif simple.

Une fois les actions de référence choisies, il faut fermer les catégories de classement. Pour cela, on procède de la manière suivante :
- on prend une action qui borne inférieurement la catégorie la plus basse,
- on prend une action qui borne supérieurement la catégorie la plus haute.

Calcul des paramètres

On calcule :
- les indices de concordance par critère,
- les indices de concordance globaux,
- les indices de discordance par critère.

Tous ces paramètres sont calculés comme dans la méthode ELECTRE III.

Les degrés de crédibilité sont calculés comme dans la méthode ELECTRE III, mais par rapport aux actions de référence.

On note b_k l'action de référence k. Le degré de crédibilité vaut alors :

$$_s(a_i, b_k) = C(a_i, b_k) \cdot \prod_{j \in \bar{F}} \frac{1 - d_j(a_i, b_k)}{1 - C(a_i, b_k)} \qquad (6.17)$$

où $C(a_i, b_k)$ est l'indice de concordance global et $\bar{F}$ est défini par :

$$\bar{F} = \{ j \mid j \in F, d_j(a_i, b_k) > C(a_i, b_k) \} \qquad (6.18)$$

et $C(a_i, b_k) \equiv C_{ik}$.

On établit maintenant les relations de surclassement. La relation de surclassement entre l'action a et l'action de référence b est établie à partir des degrés de crédibilité et d'un seuil de "coupe" constant. Le processus que l'on doit suivre pour établir la relation entre deux actions a et b est représenté à la figure 6.13.

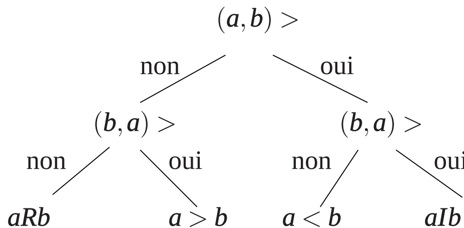


FIG. 6.13 – La relation de surclassement de la méthode ELECTRE TRI.

Avant de procéder à l'affectation des actions dans les différentes catégories, il faut vérifier sept exigences :

1. Aucune action ne peut être indifférente à plus d'une action de référence.
2. Toute action doit être attribuée à une et une seule catégorie (unicité).
3. L'affectation d'une action ne dépend pas de l'affectation d'autres actions (indépendance).
4. L'affectation des actions aux catégories doit être conforme à la définition des actions de référence (conformité).
5. Lorsque deux actions se comparent de manière identique avec les actions de référence, elles doivent être affectées à la même catégorie (homogénéité).
6. Si l'action a' domine l'action a (c'est-à-dire $\forall j, g_j(a') > g_j(a)$) alors l'action a' doit être affectée à une catégorie supérieure ou égale à celle de l'action a (monotonie).
7. Le regroupement de deux catégories voisines ne doit pas modifier l'affectation des actions aux catégories non concernées (stabilité).

On peut maintenant procéder à l'affectation des actions dans les catégories définies précédemment.

Il existe deux types de procédures d'affectation : l'affectation optimiste et l'affectation pessimiste. Celles-ci sont décrites dans le tableau 6.25.

Procédure d'affectation	Pessimiste	Optimiste
Objectif	Ranger les actions dans les catégories les plus basses possible.	Ranger les actions dans les catégories les plus hautes possible.
Procédure	Affecter l'action à une catégorie de façon telle que cette action surclasse l'action de référence basse de cette catégorie, $a S b^h \Rightarrow a \in C^{h+1}$.	Affecter l'action à une catégorie de façon telle que l'action de référence haute de cette catégorie soit préférée à l'action, $b^h > a \Rightarrow a \in C^h$.
Sens	De haut en bas.	De bas en haut.

TAB. 6.25 – Les différentes procédures d'affectation.

Ensuite, on peut représenter le résultat de ce classement sous la forme d'un graphique similaire à la figure 6.14.

Dans cette figure, on a classé les actions dans trois catégories : C_1, C_2 et C_3.

6.3.7.3 Discussion

La principale difficulté de cette méthode réside dans le fait qu'il faut pouvoir comparer chaque action avec les actions bornant les différentes catégories. Cette comparaison n'est pas toujours possible. Il est aussi assez difficile de définir des catégories *a priori*, car la définition des catégories est très liée au choix d'actions de référence.

Une application industrielle de la méthode ELECTRE TRI est présentée dans le chapitre 12.

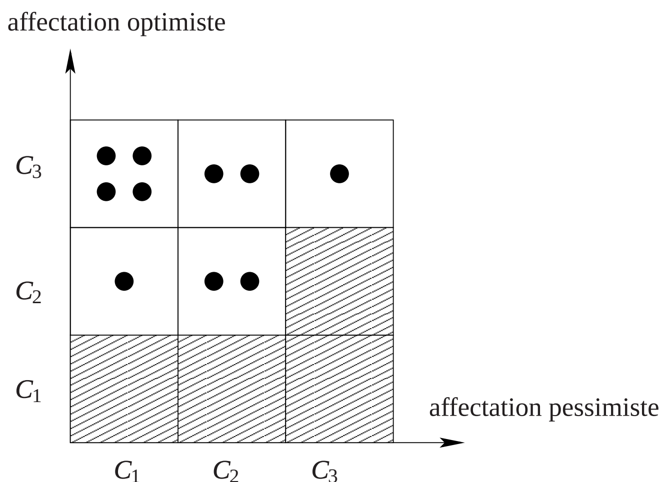


FIG. 6.14 – Résultat des deux procédures d'affectation.

6.3.8 La méthode PROMETHEE I

6.3.8.1 Principe

Cette méthode PROMETHEE (*Preference Ranking Organization Method for Enrichment Evaluations*) permet de construire un préordre partiel (il peut y avoir des ex-aequo). Elle utilise, pour cela, une relation de préférence valuée, qui donne lieu à un graphe de préférence valué. Elle a été définie par Brans, Vincke et Mareschal [Brans et al. 86] et appartient à la famille des méthodes de surclassement, définie par B. Roy.

6.3.8.2 Présentation de la méthode

La relation de préférence valuée est définie de la manière suivante :

- $P(a,b) = 0$ signifie indifférence entre l'action a et l'action b ou pas de préférence de l'action a par rapport à l'action b.
- $P(a,b) \simeq 0$ signifie une préférence faible de l'action a par rapport à l'action b.
- $P(a,b) \simeq 1$ signifie une préférence forte de l'action a par rapport à l'action b.
- $P(a,b) = 1$ signifie une préférence stricte de l'action a par rapport à l'action b.

Le lien entre la relation de préférence et un critère quelconque g_j est le suivant :

$$P_j(a,b) = G(g_j(a) - g_j(b)) \tag{6.19}$$

La fonction $G(x)$ doit être non décroissante pour $x > 0$, et égale à 0 pour $x \leq 0$. Un exemple de fonction $G(x)$ est tracé à la figure 6.15.

La fonction $g_j(x)$ est appelée "critère généralisé". Elle peut prendre plusieurs formes. Ces différentes formes sont réunies dans le tableau 6.26.

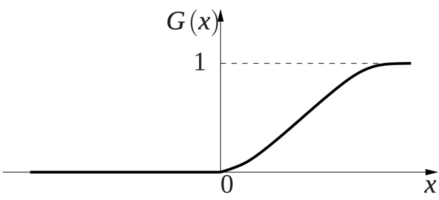


FIG. 6.15 – Exemple de fonction $G(x)$.

Type de critère généralisé	
Critère usuel	
Quasi-critère	
Critère à préférence linéaire	
Critère à niveaux	
Critère à préférence linéaire et zone d'indifférence	
Critère gaussien	

TAB. 6.26 – Différents critères généralisés.

Les différents critères généralisés ont pour équations :

– Critère usuel :

$$g_j(x) = \begin{cases} 0 & \text{si } x = 0 \\ 1 & \text{si } x \neq 0 \end{cases} \tag{6.20}$$

– Quasi-critère :

$$g_j(x) = \begin{cases} 0 & \text{si } -q \leq x \leq q \\ 1 & \text{si } x < -q \text{ ou } x > q \end{cases} \tag{6.21}$$

– Critère à préférence linéaire :

$$g_j(x) = \begin{cases} \frac{|x|}{p} & \text{si } -p \leq x \leq p \\ 1 & \text{si } x < -p \text{ ou } x > p \end{cases} \tag{6.22}$$

– Critère à niveaux :

$$g_j(x) = \begin{cases} 0 & \text{si } |x| \leq q \\ \frac{1}{2} & \text{si } q < |x| \leq p \\ 1 & \text{si } p < |x| \end{cases} \tag{6.23}$$

– Critère à préférence linéaire et zone d'indifférence :

$$g_j(x) = \begin{cases} 0 & \text{si } |x| \leq q \\ \frac{(|x|-q)}{(p-q)} & \text{si } q < |x| \leq p \\ 1 & \text{si } p < |x| \end{cases} \tag{6.24}$$

– Critère gaussien :

$$g_j(x) = 1 - exp\left(-\frac{x^2}{2 \cdot \ ^2}\right) \tag{6.25}$$

Définissons maintenant l'index de surclassement multiobjectif de l'action a par rapport à l'action b (noté $P(a,b)$) :

$$P(a,b) = \frac{\bigoplus_{i=1}^{k} \Pi_i \cdot P_i(a,b)}{\bigoplus_{i=1}^{k} \Pi_i} \tag{6.26}$$

en désignant par Π_i le poids du critère i et P_i la fonction de préférence relative au critère i. Cet index varie entre 0 et 1 :

- $P(a,b) \simeq 0$ signifie une préférence faible de l'action a par rapport à l'action b pour l'ensemble des critères ;
- $P(a,b) \simeq 1$ signifie une préférence forte de l'action a par rapport à l'action b pour l'ensemble des critères.

Nous allons nous intéresser maintenant à quelques définitions relatives à la représentation des relations de surclassement sur un graphe.

La représentation sous forme de graphe est similaire à celle des méthodes ELECTRE. La seule différence réside dans le fait que l'on indique la valeur numérique de la préférence sur les arcs. Un exemple est donné à la figure 6.16.

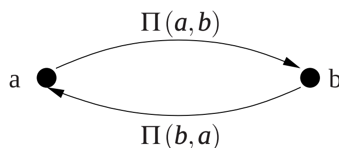


FIG. 6.16 – Graphe représentant la relation de surclassement entre a et b.

Sur un graphe de ce type, on définit :

- le flux entrant du nœud a :

$$\Phi^-(a) = \bigoplus_{b \in A} P(b,a) \tag{6.27}$$

- le flux sortant du nœud a :

$$\Phi^+(a) = \bigoplus_{b \in A} P(a,b) \tag{6.28}$$

- le flux net du nœud a :

$$\Phi(a) = \Phi^+(a) - \Phi^-(a) \tag{6.29}$$

Nous pouvons maintenant définir les deux préordres suivants :

$$\begin{cases} a\mathrm{P}^+ b & \text{si } \Phi^+(a) > \Phi^+(b) \\ aI^+ b & \text{si } \Phi^+(a) = \Phi^+(b) \end{cases} \tag{6.30}$$

$$\begin{cases} aP^- b & \text{si } \Phi^- (a) < \Phi^- (b) \\ aI^- b & \text{si } \Phi^- (a) = \Phi^- (b) \end{cases} \tag{6.31}$$

Pour finir, nous pouvons définir les préordres partiels de la méthode PROMETHEE I :
– l'action a surclasse l'action b (noté $aP_I b$) :

$$\begin{cases} si & aP^+ b & \text{et} & aP^- b \\ ou & aP^+ b & \text{et} & aI^- b \\ ou & aI^+ b & \text{et} & aP^- b \end{cases} \tag{6.32}$$

– les actions a et b sont indifférentes (noté $aI_I b$) si $aI^+ b$ et $aI^- b$;
– sinon, les actions a et b sont incomparables (noté aRb).

6.3.8.3 Discussion

La méthode PROMETHEE I fournit donc à l'utilisateur un classement des différentes actions. Le problème est que cette méthode ne permet pas de classer toutes les actions. Certaines actions peuvent rester incomparables.

La méthode PROMETHEE II permet de lever cette incomparabilité.

6.3.9 La méthode PROMETHEE II

6.3.9.1 Principe

Cette méthode permet de classer toutes les actions et ne laisse aucune action incomparable avec les autres [Brans et al. 86].

Elle permet donc de réaliser un préordre complet.

6.3.9.2 Présentation de la méthode

La seule différence avec la méthode PROMETHEE I tient dans la définition de la relation de surclassement, qui est la suivante :
– l'action a surclasse l'action b (noté $aP_{II} b$) si $\Phi(a) > \Phi(b)$,
– les actions a et b sont indifférentes si $\Phi(a) = \Phi(b)$.

6.3.9.3 Discussion

Il est plus simple pour l'utilisateur de donner une réponse à un problème de décision en utilisant un préordre complet. En revanche, le préordre partiel peut contenir plus d'informations. En effet, l'incomparabilité d'une action peut être très utile pour la prise de décision.

6.4 Bibliographie commentée

[Maystre et al. 94] Dans ce livre, on trouvera la description de toutes les méthodes d'aide à la décision ELECTRE. On trouvera, en plus de la description détaillée de chaque méthode, le traitement d'un exemple où toutes les étapes de l'application de la

méthode sont détaillées avec une grande clarté. Ce livre est une référence et toutes les notations que nous utilisons dans ce chapitre en sont inspirées.

[Scharlig 85] Le livre important pour qui cherche à être sensibilisé sur les concepts de l'aide à la décision. Ce livre a été écrit par un ancien journaliste scientifique impliqué dans l'aide à la décision. Le style de ce livre en bénéficie. Il est clair et l'auteur a réussi le tour de force de présenter tous les concepts avec un minimum d'équations. Un très bon ouvrage.

[Scharlig 96] Ce livre est une suite de [Scharlig 85]. Il décrit les méthodes ELECTRE et PROMETHEE avec la même économie d'équations.

[Roy et al. 93] Le livre du créateur de l'aide à la décision multicritère. Ce livre contient tous les éléments concernant l'aide à la décision multicritère et au-delà. Il présente aussi l'avantage de posséder plusieurs niveaux de lecture. C'est le livre que toutes les personnes utilisant l'aide à la décision se doivent de posséder.

Deuxième partie

Evaluation des méthodes et critères de choix

Chapitre 7

La mesure des performances

7.1 Introduction

Pour mesurer les performances d'une méthode d'optimisation multiobjectif, il faut disposer d' "instruments de mesure". Nous avons principalement adopté les indices de performance formulés par [Van Veldhuizen 99].

Pour illustrer chacune des formules énoncées plus loin, nous allons présenter deux exemples. Le premier exemple est issu de [Van Veldhuizen 99]. Le second illustre la progression d'une méthode dans la résolution d'un problème biobjectif (on veut minimiser les objectifs f_1 et f_2) ; nous avons représenté la surface de compromis, ainsi que l'évolution de l'ensemble des solutions sur trois générations.

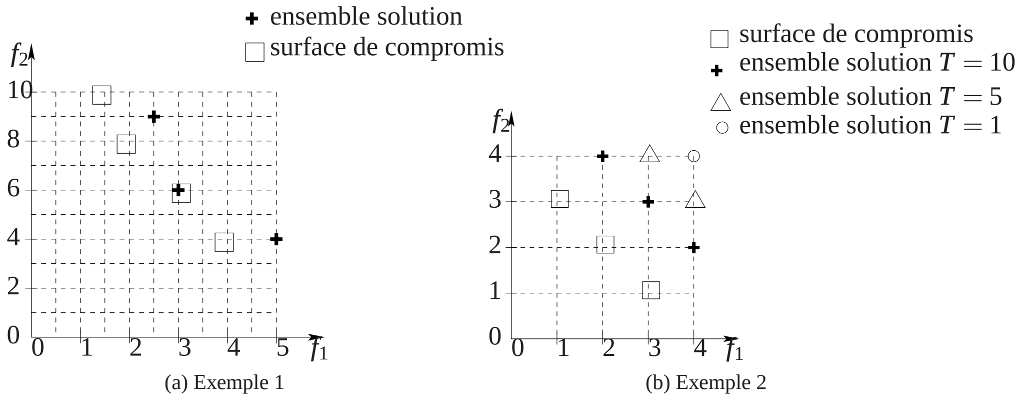


FIG. 7.1 – Les deux exemples qui illustreront les indices de performance.

A la figure 7.1-b, le paramètre T représente l'itération à laquelle a été obtenu cet ensemble de solutions.

Dans ce chapitre, nous utiliserons les notations suivantes pour définir les différents ensembles de solutions :

– La surface de compromis optimale sera notée TPF (*Theoretical Pareto Frontier* ou "surface de compromis théorique").
– L'ensemble de solutions obtenu à l'itération T sera noté $PF_{\text{current}}(T)$ (*Pareto Front Current* ou "surface de compromis courante").

– L'union des solutions non dominantes de chaque ensemble $PF_{\text{current}}(i)$, $i = 1, \cdots, T$ sera noté $PPF(T)$ (*Practical Pareto Frontier* ou "surface de compromis pratique"). La définition mathématique de cet ensemble est la suivante :

$$PPF(T) = PF_{\text{current}}(T) \bigcup PPF(T-1)$$

et

$$PPF(1) = PF_{\text{current}}(1)$$

en supprimant toutes les solutions dominées de $PPF(T)$.

Dans ce chapitre, nous allons présenter différentes grandeurs permettant de "visualiser" des caractéristiques des ensembles de solutions. Nous appellerons ces grandeurs des "métriques".

7.2 Rapport d'erreur

7.2.1 Définition

Ce rapport (*Error ratio*) permet de mesurer la non-convergence d'une méthode d'optimisation multiobjectif vers la surface de compromis. La définition de ce rapport d'erreur est la suivante :

$$E = \frac{\bigoplus_{i=1}^{n} e_i}{n} \tag{7.1}$$

avec :

n : nombre d'éléments dans l'ensemble des solutions.

e_i : $\begin{cases} = 0 & \text{si la solution } i \in \text{surface de compromis} \\ = 1 & \text{sinon} \end{cases}$

Plus cette métrique est proche de 1 moins l'ensemble des solutions a convergé vers la surface de compromis.

7.2.2 Exemples

Exemple 1

Pour cet exemple, on a :

– Pour le point de coordonnées $(2.5, 9)$: $e_1 = 1$ car ce point n'appartient pas à la surface de compromis.
– Pour le point de coordonnées $(3, 6)$: $e_2 = 0$ car ce point appartient à la surface de compromis.
– Pour le point de coordonnées $(5, 4)$: $e_3 = 1$ car ce point n'appartient pas à la surface de compromis.

L'index E vaut donc : $E = \frac{2}{3}$.

Exemple 2

Pour cet exemple, on ne calcule l'index d'erreur que pour l'ensemble des solutions obtenu pour $T = 10$. Dans ce cas, comme tous les points de cet ensemble des solutions n'appartiennent pas à la surface de compromis, on a $e_i = 1$, pour $i = 1, 2, 3$.

L'index E vaut donc : $E = 1$.

Discussion

Comme on peut le voir sur ces deux exemples, plus la métrique E est proche de 1 moins l'ensemble de solutions a convergé vers la surface de compromis.

Pour savoir si un élément appartient ou non à la surface de compromis, il suffit de réunir les deux ensembles (surface de compromis TPF + ensemble de solutions PPF), puis de ranger les éléments de ce nouvel ensemble en utilisant le classement de Pareto. Si le rang du point considéré est égal à 1, alors il appartient à la surface de compromis.

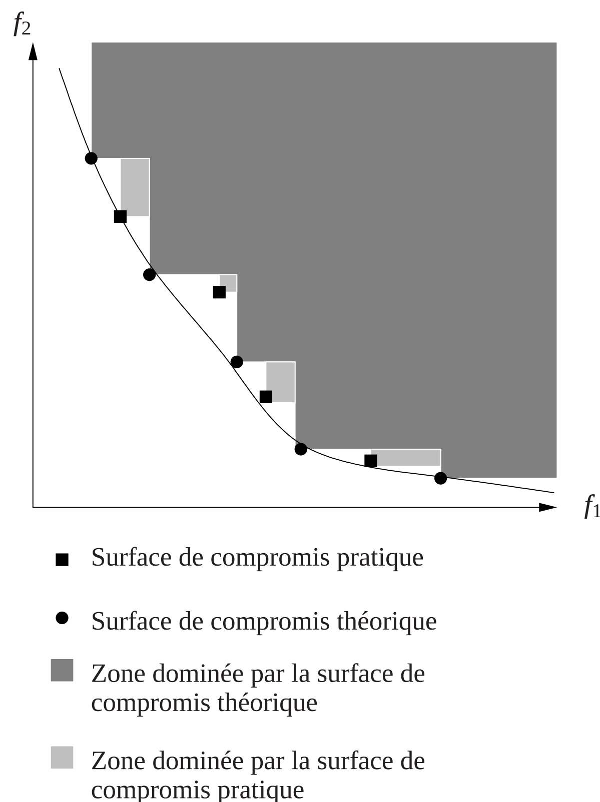


FIG. 7.2 – Exemple pour lequel le rapport d'erreur vaut 1, mais la surface de compromis pratique n'a pas pour autant convergé vers la surface de compromis théorique.

Comme on peut le voir à la figure 7.2, dans certains cas, le rapport d'erreur peut valoir 1 sans pour autant que la surface de compromis pratique ait convergé vers la surface de compromis théorique. Il suffit seulement que les solutions de la surface de compromis pratique résident dans la zone de non-domination définie par les points de la surface de compromis théorique. Ce "problème" est dû au fait que, pour pouvoir calculer le rapport d'erreur, il a

fallu “discrétiser” la surface de compromis théorique et, de cette manière, des zones de non-domination sont apparues.

7.3 Distance générationnelle

7.3.1 Définition

La distance générationnelle (*Generational Distance*) permet de mesurer à quelle distance de la surface de compromis se situe un ensemble de solutions. La définition de cette métrique est la suivante :

$$G = \frac{\left(\bigoplus_{i=1}^{n} d_i^p\right)^{\frac{1}{p}}}{n} \tag{7.2}$$

En général, on choisit $p = 2$.

On a :

d_i : distance entre la solution i et la solution la plus proche appartenant à la surface de compromis TPF ;

n : nombre d’éléments dans l’ensemble de solutions PPF.

7.3.2 Exemples

Exemple 1

Appliquons la définition aux éléments de l’ensemble de solutions :

$G = \frac{\left[\left[(2.5-2)^2+(9-8)^2\right]+\left[(3-3)^2+(6-6)^2\right]+\left[(5-4)^2+(4-4)^2\right]\right]^{\frac{1}{2}}}{3} = 0.5$

Exemple 2

Pour cet exemple, on calcule la métrique pour les trois ensembles de solutions ($T = 1$, $T = 5$, $T = 10$) :

– Pour $T = 1$, on a :
  $G_1 = \frac{\left[(4-2)^2+(4-2)^2\right]^{\frac{1}{2}}}{1} = 2.82$
– Pour $T = 5$, on a :
  $G_2 = \frac{\left[\left[(4-2)^2+(3-2)^2\right]+\left[(3-2)^2+(4-2)^2\right]\right]^{\frac{1}{2}}}{2} = 1.58$
– Pour $T = 10$, on a :
  $G_3 = \frac{\left[\left[(3-4)^2+(1-2)^2\right]+\left[(2-3)^2+(2-3)^2\right]+\left[(1-2)^2+(3-4)^2\right]\right]^{\frac{1}{2}}}{3} = 0.81$

Discussion

Comme le montre l’exemple 2, plus l’ensemble de solutions est éloigné de la surface de compromis, plus la valeur de la distance générationnelle est grande.

Cette métrique pourrait donc être utilisée comme critère d’arrêt de la méthode d’optimisation, à condition de connaître la surface de compromis TPF par avance (ce qui est le cas

uniquement lorsque l'on travaille avec un problème test multiobjectif). On peut aussi utiliser la dérivée de cette métrique comme critère d'arrêt. En effet, si la variation de G est très faible, itération après itération, cela voudra dire que l'ensemble de solutions *PPF* ne se rapproche plus de la surface de compromis optimale. Cette circonstance peut se produire lorsque l'ensemble de solutions a convergé vers une surface de compromis fantôme, différente de la surface de compromis optimale.

Il existe une version modifiée de la distance générationnelle :

$$GD = \frac{\bigoplus_{i=1}^{n} d_i}{n} \tag{7.3}$$

GD correspond à une distance moyenne par rapport à la surface de compromis. Elle est exploitée dans la métrique présentée dans la section suivante.

7.4 Métrique STDGD

7.4.1 Définition

Cette métrique (*Standard Deviation from the Generational Distance*) permet de mesurer la déformation de la surface de compromis calculée par rapport à la surface de compromis théorique. La définition est la suivante :

$$STDGD = \frac{\bigoplus_{i=1}^{n} (d_i - GD)^2}{n} \tag{7.4}$$

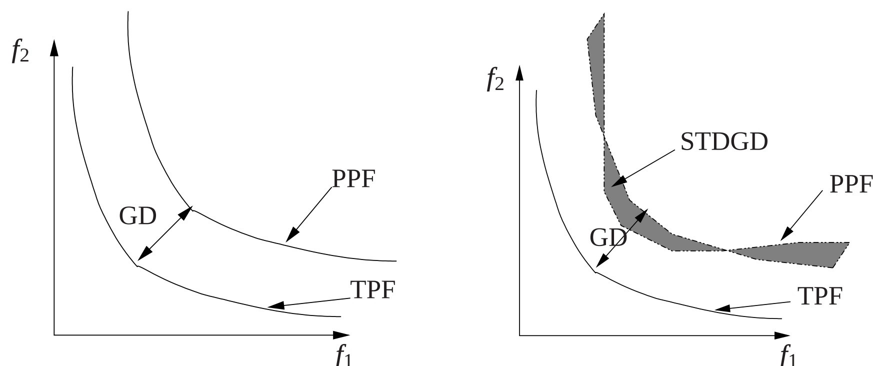


FIG. 7.3 – La définition de la métrique STDGD.

Le principe de cette métrique est illustré à la figure 7.3 où :

PPF : désigne la surface de compromis calculée ;

TPF : désigne la surface de compromis théorique ;

GD : désigne la distance générationnelle.

La zone grisée désigne la surface que calcule la métrique STDGD.

7.4.2 Discussion

Cette métrique calcule l'erreur au sens des moindres carrés entre la surface de compromis théorique décalée de la distance GD et la surface de compromis pratique. Cette mesure traduit donc la déformation de la surface de compromis pratique.

Cette métrique a été conçue dans le but de distinguer les méthodes qui calculent des surfaces de compromis déformées par rapport à la surface de compromis théorique, des méthodes qui calculent des surfaces de compromis respectant la forme théorique.

7.5 Erreur maximale à la surface de compromis

7.5.1 Définition

Cette métrique (*Maximum Pareto Front Error*) permet de mesurer la distance entre la surface de compromis et l'ensemble de solutions. Sa définition est la suivante (pour un problème à deux objectifs) :

$$ME = \max_j \left(\min_i \left(\left| f_1^i(\overrightarrow{x}) - f_1^j(\overrightarrow{x}) \right|^p + \left| f_2^i(\overrightarrow{x}) - f_2^j(\overrightarrow{x}) \right|^p \right)^{\frac{1}{p}} \right) \tag{7.5}$$

Il s'agit, en fait, de la plus grande distance minimale entre les éléments de l'ensemble de solutions et leurs voisins les plus proches appartenant à la surface de compromis.

7.5.2 Exemples

Exemple 1

Pour trouver les plus grandes distances minimales, on considère chaque point de l'ensemble solution et on calcule les différentes distances par rapport aux points de la surface de compromis (voir tableau 7.1).

– Pour le point de l'ensemble de solutions de coordonnées $(2.5, 9)$, la distance minimale vaut 1.12 ;
– Pour le point de l'ensemble de solutions de coordonnées $(3, 6)$, la distance minimale vaut 0 ;
– Pour le point de l'ensemble de solutions de coordonnées $(5, 4)$, la distance minimale vaut 1.

La plus grande distance minimale vaut 1.12 donc, $ME = 1.12$.

Exemple 2

Nous suivons la même démarche que pour l'exemple 1.

Les plus grandes distances minimales valent :

– pour l'ensemble de solutions avec $T = 10$ (voir tableau 7.2), $ME = 1.41$;
– pour l'ensemble de solutions avec $T = 5$ (voir tableau 7.3), $ME = 2.23$;

Point de l'ensemble solution	Point de la surface de compromis	Distance
$(2.5,9)$	$(1.5,10)$	1.41
$(2.5,9)$	$(2,8)$	1.12
$(2.5,9)$	$(3,6)$	3.04
$(2.5,9)$	$(4,4)$	5.22
$(3,6)$	$(1.5,10)$	4.27
$(3,6)$	$(2,8)$	2.23
$(3,6)$	$(3,6)$	0
$(3,6)$	$(4,4)$	2.23
$(5,4)$	$(1.5,10)$	6.94
$(5,4)$	$(2,8)$	5
$(5,4)$	$(3,6)$	2.82
$(5,4)$	$(4,4)$	1

TAB. 7.1 – Exemple 1 : calcul de l'erreur maximale à la surface de compromis.

Point de l'ensemble de solutions	Point de la surface de compromis	Distance
$(4,2)$	$(3,1)$	1.41
$(4,2)$	$(2,2)$	2
$(4,2)$	$(1,3)$	3.16
$(3,3)$	$(3,1)$	2
$(3,3)$	$(2,2)$	1.41
$(3,3)$	$(1,3)$	2
$(2,4)$	$(3,1)$	3.16
$(2,4)$	$(2,2)$	2
$(2,4)$	$(1,3)$	1.41

TAB. 7.2 – Exemple 2 (pour $T = 10$) : calcul de l'erreur maximale à la surface de compromis.

Point de l'ensemble de solutions	Point de la surface de compromis	Distance
$(4,3)$	$(3,1)$	2.23
$(4,3)$	$(2,2)$	2.23
$(4,3)$	$(1,3)$	3
$(3,4)$	$(3,1)$	3
$(3,4)$	$(2,2)$	2.23
$(3,4)$	$(1,3)$	2.23

TAB. 7.3 – Exemple 2 (pour $T = 5$) : calcul de l'erreur maximale à la surface de compromis.

– pour l'ensemble de solutions avec $T = 1$ (voir tableau 7.4), $ME = 2.82$.

Point de l'ensemble de solutions	Point de la surface de compromis	Distance
$(4,4)$	$(3,1)$	3.16
$(4,4)$	$(2,2)$	2.82
$(4,4)$	$(1,3)$	3.16

TAB. 7.4 – Exemple 2 (pour $T = 1$) : calcul de l'erreur maximale à la surface de compromis.

Discussion

Comme le montre l'exemple 2, plus l'ensemble de solutions est "éloigné" de la surface de compromis TPF, plus la métrique ME est grande. Cette métrique mesure donc la distance entre la surface de compromis TPF et l'ensemble de solutions PPF.

7.6 Hypersurface et rapport d'hypersurface

7.6.1 Définition

Cette métrique (*Hyperarea and Ratio*) permet de mesurer la surface occupée par l'ensemble de solutions PPF. Sa définition est la suivante :

$$H = \left\{ \bigcup_i a_i \,|\, v_i \in PPF \right\} \tag{7.6}$$

avec :

a_i : surface occupée par la solution v_i (se reporter à la figure 7.4 pour une illustration de cette surface).

Une façon de calculer cette métrique dans le cas d'un ensemble de solutions PPF en deux dimensions est la suivante :

Etape 1 : Extraire la surface de compromis de l'ensemble de solutions PPF. On appelle S_c cette surface de compromis.

Etape 2 : Ranger les éléments de S_c par ordre croissant de la fonction objectif f_1.

Etape 3 : Appliquer l'algorithme suivant :

1. $i = 0$, $H = f_1(x_0) \cdot f_2(x_0)$
2. $i = i + 1$
3. si $i \neq N$, $H = H + [f_1(x_i) - f_1(x_{i-1})] \cdot f_2(x_i)$
   allez en 2
4. si $i = N$ alors fin

avec :

N : nombre d'éléments dans l'ensemble de solutions S_c,

x_i : un élément de l'ensemble de solutions S_c.

On peut aussi définir un rapport d'hypersurface :

$$HR = \frac{H_{es}}{H_{sc}} \quad (7.7)$$

avec :

H_{es} : surface occupée par l'ensemble de solutions PPF ;

H_{sc} : surface occupée par la surface de compromis TPF.

7.6.2 Exemples

Exemple 1

Pour cet exemple, on calcule les deux surfaces.

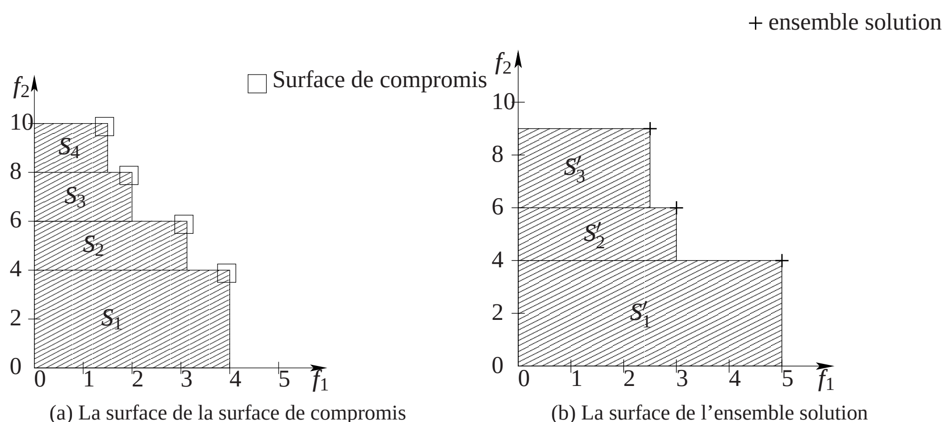


FIG. 7.4 – Le calcul des surfaces.

Pour la surface de compromis, on a :
$H_{es} = S_1 + S_2 + S_4$
$H_{es} = 5 \cdot 4 + 3 \cdot (6-4) + 2.5 \cdot (9-6)$
$H_{sc} = S_1 + S_2 + S_3 + S_4$
$H_{sc} = 4 \cdot 4 + 3 \cdot (6-4) + 2 \cdot (8-6) + 1.5 \cdot (10-8)$
donc :
$H_{es} = 33.5$
$H_{sc} = 29$
On calcule le rapport d'hypersurface :
$HR = \frac{H_{es}}{H_{sc}} = \frac{33.5}{29} = 1.155$

Exemple 2

On calcule les surfaces pour les trois ensembles de solutions, ainsi que pour la surface de compromis.

– Pour la surface de compromis :
  $H_{sc} = 3 \cdot 1 + 2 \cdot (2-1) + 1 \cdot (3-2)$
  $H_{sc} = 6$

- Pour l'ensemble de solutions tel que $T = 10$:
  $H_{es_1} = 4 \cdot 2 + 3 \cdot (3-2) + 2 \cdot (4-3)$
  $H_{es_1} = 13$
  On a donc le rapport d'hypersurface suivant :
  $HR_1 = \frac{H_{es_1}}{H_{sc}} = \frac{13}{6} = 2.17$
- Pour l'ensemble de solutions tel que $T = 5$:
  $H_{es_2} = 4 \cdot 3 + 3 \cdot (4-3)$
  $H_{es_2} = 15$
  On a donc le rapport d'hypersurface suivant :
  $HR_2 = \frac{H_{es_2}}{H_{sc}} = \frac{15}{6} = 2.5$
- Pour l'ensemble de solutions tel que $T = 1$:
  $H_{es_3} = 4 \cdot 4$
  $H_{es_3} = 16$
  On a donc le rapport d'hypersurface suivant :
  $HR_3 = \frac{H_{es_3}}{H_{sc}} = \frac{16}{6} = 2.67$

Discussion

Comme les métriques “erreur maximale à la surface de compromis” et “distance générationnelle”, le rapport d'hypersurface permet de mesurer la convergence de l'ensemble de solutions PPF vers la surface de compromis TPF. Plus la valeur de cette métrique est proche de 1, meilleure est la surface de compromis.

- Si la valeur du rapport d'hypersurface est inférieure à 1, la surface occupée sous la surface de compromis pratique est inférieure à la surface occupée sous la surface de compromis théorique. On peut dire que les points de la surface de compromis pratique ne se répartissent pas sur toute la surface de compromis.
- Si la valeur du rapport d'hypersurface est supérieure à 1, la surface de compromis pratique est plutôt éloignée de la surface de compromis théorique. On n'a pas suffisamment convergé pour que cette métrique nous donne une information quant à la répartition des points sur la surface de compromis.

7.7 Espacement

7.7.1 Définition

La définition de la métrique d'espacement (*Spacing*) est la suivante :

$$S = \left[\frac{1}{n-1} \cdot \bigoplus_{i=1}^{n} \left(\overline{d} - d_i \right)^2 \right]^{\frac{1}{2}} \tag{7.8}$$

avec

$$d_i = \min_j \left(\left| f_1^i(\overrightarrow{x}) - f_1^j(\overrightarrow{x}) \right| + \left| f_2^i(\overrightarrow{x}) - f_2^j(\overrightarrow{x}) \right| \right) \tag{7.9}$$

$i, j = 1, \cdots, n$ et $i \neq j$.
et

$\overline{d}$: moyenne de tous les d_i ;

n : nombre d'éléments dans l'ensemble solution.

Cette métrique permet de mesurer l'uniformité de la répartition des points de l'ensemble solution dans le plan (f_1, f_2).

7.7.2 Exemples

Exemple 1

Point i	Point j	Distance
$(2.5,9)$	$(3,6)$	3.5
$(2.5,9)$	$(5,4)$	7.5
$(3,6)$	$(2.5,9)$	3.5
$(3,6)$	$(5,4)$	4
$(5,4)$	$(2.5,9)$	7.5
$(5,4)$	$(3,6)$	4

Pour le point de coordonnées $(2.5,9)$, $d_1 = 3.5$.
Pour le point de coordonnées $(3,6)$, $d_2 = 3.5$.
Pour le point de coordonnées $(5,4)$, $d_3 = 4$.
On a donc :
$\bar{d} = \frac{3.5+3.5+4}{3} = 3.67$
On calcule, pour finir, l'index S :
$$S = \left[\frac{1}{2} \cdot \left((3.67-3.5)^2 + (3.67-3.5)^2 + (3.67-4)^2\right)\right]^{\frac{1}{2}}$$
$S = 0.288$

Exemple 2

Nous suivons la même démarche pour l'exemple 2.
Pour l'ensemble solution tel que $T = 10$, on a :

Point i	Point j	Distance
$(4,2)$	$(3,3)$	2
$(4,2)$	$(2,4)$	4
$(3,3)$	$(4,2)$	2
$(3,3)$	$(2,4)$	2
$(2,4)$	$(4,2)$	4
$(2,4)$	$(3,3)$	2

Pour le point de coordonnées $(4,2)$, $d_1 = 2$.
Pour le point de coordonnées $(3,3)$, $d_2 = 2$.
Pour le point de coordonnées $(2,4)$, $d_3 = 2$.
On a donc $\bar{d} = 2$ et $S = 0$.

Il n'est pas intéressant de calculer cet index pour l'ensemble de solutions tel que $T = 5$ car il n'y a que deux points, ils seront donc forcément bien espacés. Pour l'ensemble de solutions tel que $T = 1$, le calcul de cet index n'a pas de sens.

Discussion

Comme le montrent ces deux exemples, la métrique d'espacement permet de mesurer l'uniformité de la répartition des solutions dans un ensemble (PPF ou TPF). Plus celle-ci est proche de 0, mieux les points sont répartis.

Dans certains cas, cette métrique peut être difficile à interpréter. En effet, quand on travaille sur un problème test discontinu, la surface de compromis de ce problème possède déjà des trous (discontinuité introduite par la forme de la surface de compromis). Ces trous vont augmenter la valeur calculée par cette métrique. Il vaut mieux éviter d'utiliser cette métrique sur un problème test discontinu.

7.8 Métrique HRS

7.8.1 Définition

La métrique *Spacing* calculant une erreur moyenne par rapport à un espacement idéal, elle a tendance à “dissimuler” les écarts importants. C'est pour cela que nous avons conçu la métrique HRS (*Hole Relative Size*). Elle permet de mesurer la taille du plus grand trou dans l'espacement des points sur la surface de compromis. La définition de la métrique HRS est la suivante :

$$HRS = \frac{\max_i d_i}{\overline{d}} \tag{7.10}$$

où $\overline{d}$ et d_i ont la même signification que pour la métrique *Spacing*.

7.9 Ensemble des vecteurs non dominés ultimes

7.9.1 Définition

On crée un ensemble qui contient les meilleures solutions non dominées obtenues après chaque génération (l'ensemble PPF donc). On définit ensuite une métrique (*Overall Nondominated Vector Generation*) qui est le cardinal de cet ensemble :

$$ONVG = \text{card}\left(PPF\right) \tag{7.11}$$

On peut ensuite comparer le cardinal de l'ensemble de solutions PPF avec le cardinal de la surface de compromis TPF, en formant le rapport :

$$ONVGR = \frac{\text{card}\left(PPF\right)}{\text{card}\left(TPF\right)} \tag{7.12}$$

7.9.2 Exemples

Exemple 1

Pour cet exemple, la surface de compromis TPF est composée de quatre solutions non dominées et l'ensemble de solutions PPF est composé de trois solutions non dominées. On a donc :
$ONVG = 3$
$ONVGR = \frac{3}{4} = 0.75$

Exemple 2

- Pour l'ensemble de solutions tel que $T = 10$, on a :
  $ONVG_1 = 3$
  $ONVGR_1 = \frac{3}{3} = 1$
- Pour l'ensemble de solutions tel que $T = 5$, on a :
  $ONVG_2 = 2$
  $ONVGR_2 = \frac{2}{3} = 0.66$
- Pour l'ensemble de solutions tel que $T = 1$, on a :
  $ONVG_1 = 1$
  $ONVGR = \frac{1}{3} = 0.33$

Discussion

Le rapport de l'ensemble des vecteurs non dominés ultimes permet de connaître la proportion de solutions non dominées existant dans l'ensemble de solutions PPF par rapport à la surface de compromis TPF.

7.10 Mesure de la progression

7.10.1 Définition

On mesure l'amélioration de la fonction objectif f_i en utilisant la métrique suivante (*Progress Measure*) :

$$P = \ln \sqrt{\frac{f_{i_{max}}(0)}{f_{i_{max}}(k)}} \tag{7.13}$$

en désignant par $f_{i_{max}}(k)$ la meilleure valeur de la fonction objectif i à l'itération k.

Pour prendre en compte la possibilité de l'existence de plusieurs solutions, Van Veldhuizen modifie cette expression de la manière suivante [Van Veldhuizen 99] :

$$RP = \ln \sqrt{\frac{G_0}{G_k}} \tag{7.14}$$

où G_k est la distance générationnelle à l'itération k.

7.10.2 Exemple

Exemple 2

- Pour l'ensemble solution *PPF* tel que $T = 1$, on a :
  $f_{2_{max}}(1) = 4$ et $f_{1_{max}}(1) = 4$.
  Ces deux valeurs vont servir de point de départ pour le calcul de la métrique de progression.
- Pour l'ensemble de solutions *PPF* tel que $T = 5$, on a :
  $f_{2_{max}}(5) = 3$ et $f_{1_{max}}(5) = 3$.

  On a alors :

  $P_{11} = \ln\sqrt{\frac{f_{1_{max}}(1)}{f_{1_{max}}(5)}} = \ln\sqrt{\frac{4}{3}} = 0.14$

  $P_{21} = \ln\sqrt{\frac{f_{2_{max}}(1)}{f_{2_{max}}(5)}} = \ln\sqrt{\frac{4}{3}} = P_{11}$
- Pour l'ensemble de solutions *PPF* tel que $T = 10$, on a :
  $f_{2_{max}}(10) = 2$ et $f_{1_{max}}(10) = 2$.

  On a alors :

  $P_{12} = \ln\sqrt{\frac{f_{1_{max}}(1)}{f_{1_{max}}(10)}} = \ln\sqrt{\frac{4}{2}} = 0.34$

  $P_{22} = \ln\sqrt{\frac{f_{2_{max}}(1)}{f_{2_{max}}(10)}} = \ln\sqrt{\frac{4}{2}} = P_{12}$

Discussion

Cette métrique permet de mesurer la progression de la méthode d'optimisation multiobjectif vers la solution “idéale”.

7.11 Génération de vecteurs non dominés

Cette métrique (*Generational Nondominated Vector Generation*) permet de mesurer combien de solutions non dominées sont générées à chaque étape.

$$GNVG = \text{card}\left(PF_{\text{current}}(i)\right) \tag{7.15}$$

où $PF_{\text{current}}(i)$ est l'ensemble de solutions à l'itération i.

7.12 Ajout de vecteurs non dominés

7.12.1 Définition

Cette métrique permet de mesurer combien de nouveaux vecteurs non dominés ont été ajoutés entre deux itérations. La définition de cette métrique est la suivante :

$$NVA = \text{card}\left(PPF(i)\right) - \text{card}\left(PPF(i-1)\right) \tag{7.16}$$

7.12.2 Exemple

Exemple 2

– Pour l'ensemble de solutions tel que $T = 1$, on a $\text{card}(PPF(1)) = 1$.
– Pour l'ensemble de solutions tel que $T = 5$, on a $\text{card}(PPF(5)) = 2$.
  Donc, $NVA_1 = \text{card}(PPF(5)) - \text{card}(PPF(1)) = 1$.
  Il y a eu ajout d'une solution non dominée entre l'itération 1 et l'itération 5.
– Pour l'ensemble de solutions tel que $T = 10$, on a $\text{card}(PPF(10)) = 3$.
  Donc, $NVA_2 = \text{card}(PPF(10)) - \text{card}(PPF(5)) = 1$.
  Il y a eu ajout d'une solution non dominée entre l'itération 5 et l'itération 10.

Discussion

La métrique *NVA* peut, dans certains cas, être négative. En effet, la méthode d'optimisation peut perdre des solutions non dominées au cours de sa progression. Cette métrique peut servir à arrêter une méthode d'optimisation. Si la métrique stagne sur la valeur 0 depuis un certain nombre d'itérations, cela veut dire qu'elle n'est plus capable de trouver de nouvelles solutions non dominées. Cependant, il ne faut pas oublier qu'une méthode peut ne plus créer de nouvelles solutions, mais être encore capable de faire progresser les solutions existantes vers la surface de compromis optimale TPF.

Cette métrique peut être vue comme un "taux" (elle ne procure pas des valeurs comprises absolument entre 0 et 1) de production de solutions non dominées.

7.13 Vagues

Cette métrique (*Waves*) correspond au rang maximal de la classification de Pareto. Ce rang peut être obtenu en utilisant l'algorithme 7.1.

Algorithm 7.1 MaxRangPareto.

```
S contient l'ensemble des solutions
d = 0
while card(S) ≠ 0
  S = S − Pareto(S)
  d = d + 1
end while
MaxRangPareto = d
```

$$W = MaxRangPareto(PPF) \tag{7.17}$$

Dans l'algorithme 7.1, la fonction Pareto détermine un ensemble contenant les solutions non dominées de l'ensemble donné en paramètre.

W permet de mesurer l'étalement de l'ensemble de solutions PPF. Si W est grand, cela signifie que la population n'est pas regroupée au niveau de la surface de compromis TPF : beaucoup d'individus ne sont pas "efficaces".

Si la valeur de cette métrique se rapproche de 1, cela signifie que les individus tendent à converger vers la surface de compromis. En général, cette métrique est utilisée au cours d'une

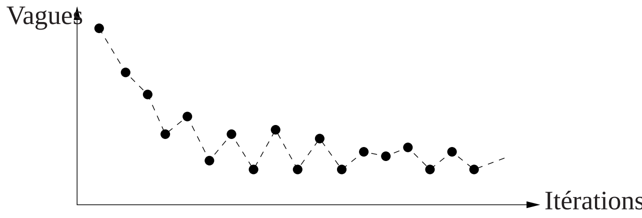


FIG. 7.5 – La métrique W au cours d'une optimisation.

optimisation afin de visualiser la convergence de l'ensemble vers la surface de compromis. L'allure de la courbe ainsi obtenue est celle de vagues (voir figure 7.5).

Cette métrique mesure l'épaisseur de l'ensemble de solutions *PPF* en nombre de surfaces de compromis que comporte cet ensemble.

7.14 Métriques de Zitzler

Dans [Zitzler et al. 99], plusieurs métriques de performance sont présentées. Ces métriques peuvent se répartir en deux familles :

- les métriques relatives : elles permettent de comparer deux surfaces de compromis ;
- les métriques absolues : elles permettent de mesurer la "qualité" d'une surface de compromis.

Les métriques présentées jusqu'à présent sont des métriques absolues, car elles ne permettent pas de comparer deux surfaces de compromis pratiques.

7.14.1 Les métriques relatives

Dans [Zitzler et al. 99], seule une métrique relative est présentée. La définition de cette métrique est donnée ci-dessous.

Soient F' et F'' deux surfaces de compromis. La fonction C suivante transforme les deux surfaces de compromis F' et F'' en un réel compris dans l'intervalle $[0,1]$:

$$C\left(F',F''\right) = \frac{\text{card}\left(\left\{\overrightarrow{x''} \in F'' ; \exists \overrightarrow{x'} \in F' \mid \overrightarrow{x'} \preceq \overrightarrow{x''}\right\}\right)}{\text{card}\left(F''\right)} \tag{7.18}$$

où $\text{card}\,(F)$ désigne le nombre d'éléments contenus dans l'ensemble F et $\overrightarrow{x'} \preceq \overrightarrow{x''}$ signifie que le vecteur $\overrightarrow{x'}$ domine le (ou est équivalent au) vecteur $\overrightarrow{x''}$.

Quand $C\left(F',F''\right) = 1$, cela signifie que toutes les solutions de l'ensemble F' sont dominées par (ou égales à) des solutions de l'ensemble F''.

Quand $C\left(F',F''\right) = 0$, cela signifie que toutes les solutions de l'ensemble F' dominent (au sens large) les solutions de l'ensemble F''.

Il faut toujours calculer $C\left(F',F''\right)$ et $C\left(F'',F'\right)$ car $C\left(F',F''\right)$ n'est pas nécessairement égale à $1-C\left(F'',F'\right)$.

Cette métrique permet donc de calculer quelle portion de la surface de compromis F' domine la surface de compromis F''.

Exemple 1

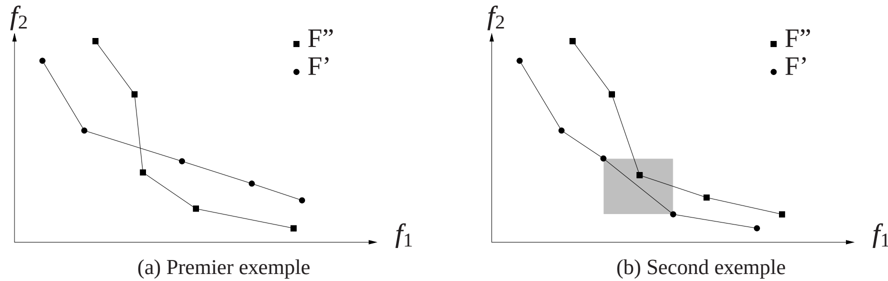


FIG. 7.6 – Exemples d’application de la métrique relative de Zitzler.

A la figure 7.6-a, on peut voir deux ensembles F' et F'' que nous désirons comparer en utilisant la métrique relative de Zitzler.

Nous allons commencer par calculer $C(F', F'')$:

$\mathrm{card}(F'')$ correspond au nombre de solutions contenues dans l’ensemble F'' : $\mathrm{card}(F'') = 5$.

Le numérateur de l’expression nous donne le nombre de solutions de l’ensemble F'' qui sont dominées par des solutions de l’ensemble F'. Dans notre exemple, on a :

$$\mathrm{card}\left(\left\{\overrightarrow{x''} \in F''; \exists \overrightarrow{x'} \in F' \,|\, \overrightarrow{x'} \preceq \overrightarrow{x''}\right\}\right) = 2$$

car deux solutions de l’ensemble F'' sont dominées par des solutions de l’ensemble F'.

Donc, $C(F', F'') = \frac{2}{5} = 0.4$.

Calculons maintenant $C(F'', F')$:

$\mathrm{card}(F') = 5$.

$\mathrm{card}\left(\left\{\overrightarrow{x'} \in F'; \exists \overrightarrow{x''} \in F'' \,|\, \overrightarrow{x''} \preceq \overrightarrow{x'}\right\}\right) = 3$.

Donc, $C(F'', F') = \frac{3}{5} = 0.6$.

On obtient la matrice suivante :

C(colonne, ligne)	F'	F''
F'	$\times$	0.4
F''	0.6	$\times$

Exemple 2

Nous allons maintenant traiter les ensembles de la figure 7.6-b.

Nous allons commencer par calculer $C(F', F'')$:

$\mathrm{card}(F'') = 5$.

$\mathrm{card}\left(\left\{\overrightarrow{x''} \in F''; \exists \overrightarrow{x'} \in F' \,|\, \overrightarrow{x'} \preceq \overrightarrow{x''}\right\}\right) = 4$.

Donc, $C(F', F'') = \frac{4}{5} = 0.8$.

Calculons maintenant $C(F'', F')$:

$\mathrm{card}(F') = 5$.

$\mathrm{card}\left(\left\{\overrightarrow{x'} \in F'; \exists \overrightarrow{x''} \in F'' \,|\, \overrightarrow{x''} \preceq \overrightarrow{x'}\right\}\right) = 0$.

Donc, $C(F'', F') = \frac{0}{5} = 0$.

On obtient la matrice suivante :

C (colonne, ligne)	F'	F''
F'	$\times$	0.8
F''	0	$\times$

Discussion

Comme on peut le remarquer sur l'exemple 2, on n'a pas nécessairement $C(F', F'') + C(F'', F') = 1$. L'expression $1 - C(F', F'') - C(F'', F')$ mesure le nombre de solutions des ensembles F' et F'' qui sont non dominées entre elles. Le résultat calculé par la métrique relative de Zitzler est une image de la proportion de la surface de compromis d'un ensemble F' qui est dominée par la surface de compromis d'un ensemble F''.

7.14.2 Les métriques absolues

Dans [Zitzler et al. 99], trois métriques absolues sont présentées. La première est strictement équivalente à la distance générationnelle présentée dans la section 7.3. Elle a pour définition :

$$M_1(F') = \frac{1}{\mathrm{card}(F')} \cdot \bigoplus_{\overrightarrow{x'} \in F'} \min\left(\left\|\overrightarrow{x'} - \overrightarrow{x''}\right\|; \overrightarrow{x''} \in F''\right) \qquad (7.19)$$

avec

F' surface de compromis pratique ;

F'' surface de compromis théorique ;

$\mathrm{card}(F)$ nombre d'éléments dans l'ensemble F ;

$\left\|\overrightarrow{x'} - \overrightarrow{x''}\right\|$ distance (euclidienne par exemple) entre le vecteur $\overrightarrow{x'}$ et le vecteur $\overrightarrow{x''}$.

Cette métrique mesure la distance moyenne entre une surface de compromis pratique (F' ou surface de compromis que l'on vient de calculer) et une surface de compromis théorique (F'').

Une seconde métrique est présentée. Elle permet de calculer un index image du nombre de niches d'une certaine taille que l'on peut trouver dans une surface de compromis. La définition de cette métrique est la suivante :

$$M_2(F', F'') = \frac{1}{\mathrm{card}(F') - 1} \cdot \bigoplus_{\overrightarrow{x'} \in F'} \mathrm{card}\left(\left\{\overrightarrow{y'} \in F' \mid \left\|\overrightarrow{x'} - \overrightarrow{y'}\right\| > \quad \right\}\right) \qquad (7.20)$$

Cette métrique a une valeur comprise dans l'intervalle $[0, \mathrm{card}(F')]$.

Quand $M_2(F', F'') = \mathrm{card}(F')$, cela signifie qu'aucun vecteur de F' n'a de voisin situé à une distance inférieure à .

Il s'agit donc d'une métrique d'espacement. La différence entre cette métrique M_2 et la métrique d'espacement décrite dans la section 7.7 est que M_2 procure une valeur "normalisée" (il suffit de diviser le résultat de la métrique M_2 par $\text{card}\left(F'\right)$ pour obtenir une mesure comprise entre 0 et 1).

Pour finir, une dernière métrique est proposée, qui mesure l'étendue de la surface de compromis. Sa définition est la suivante :

$$M_3\left(F'\right) = \sqrt{\bigoplus_{i=1}^{n} \max\left(\left\|x'_i - y'_i\right\|, \overrightarrow{x'}, \overrightarrow{y'} \in F'\right)} \tag{7.21}$$

où n désigne la dimension de l'espace F'.

Cette métrique mesure donc l'hypervolume qui contient la surface de compromis (somme des plus grandes distances sur chaque composante i).

7.15 Métrique de Laumanns

Une mesure possible de la taille de l'espace dominé des fonctions objectif est la suivante :

Définition :
Soit un vecteur de fonctions objectif arbitraire $\overrightarrow{f}$.
Soit $\overline{\overrightarrow{f}} = \left(\overline{f_1}, \overline{f_2}, \cdots, \overline{f_N}\right)$ le vecteur des composantes minimum de $\overrightarrow{f}$.
Soit $\underline{\overrightarrow{f}} = \left(\underline{f_1}, \underline{f_2}, \cdots, \underline{f_N}\right)$ le vecteur des composantes maximum de $\overrightarrow{f}$.
On note :

$$\begin{aligned} \left(\overrightarrow{f}\right) \quad = \quad & \left\{\overrightarrow{f} \in \mathbb{R}^N \mid \right. \\ & \overrightarrow{f} = \underline{\overrightarrow{f}} + \oplus_{i=1}^{N} a_i \cdot \left(\left(\underline{f_1}, \cdots, \overline{f_i}, \cdots, \underline{f_N}\right) - \underline{\overrightarrow{f}}\right) \\ & \left. a_i \in [0,1]\right\} \end{aligned} \tag{7.22}$$

et :

$$D\left(A^*, \overrightarrow{f}\right) = \bigcup_{\overrightarrow{f} = F\left(\overrightarrow{x}\right) \in A^*} \left\{\overrightarrow{f'} = F\left(\overrightarrow{x'}\right) \in \quad \mid \overrightarrow{x} \prec \overrightarrow{x'}\right\} \tag{7.23}$$

La mesure de la taille de l'espace dominé est alors :

$$s\left(A^*, F\right) = \frac{(D)}{(\quad)} \tag{7.24}$$

où $\quad (A)$ est la mesure de Lebesgue de l'ensemble fermé A.

La mesure de Lebesgue permet de mesurer la taille des ensembles, même si ces ensembles sont constitués d'une quantité infinie d'éléments. Intuitivement, cette mesure correspond à l'hypervolume de cet ensemble (volume dans un espace à m dimensions, où m correspond au nombre d'objectifs de notre problème d'optimisation, par exemple).

Cette métrique est présentée dans [Laumanns et al. 00] et se rapproche de la troisième métrique de Zitzler. La mesure obtenue a une valeur comprise entre 0 et 1.

7.16 Bibliographie commentée

[Van Veldhuizen 99] Cette thèse fait l'inventaire d'une grande partie des métriques utilisées pour la mesure de performances des méthodes d'optimisation multiobjectif.

[Zitzler et al. 99] Dans cet article est présentée une métrique de performance qui est, depuis, très utilisée pour la comparaison de méthodes d'optimisation.

[Cooper et al. 00] C'est un livre sur l'analyse d'enveloppe de données. Ce domaine s'intéresse à la mesure de performances de systèmes économiques en utilisant la méthode DEA. C'est un domaine assez récent, mais dont les méthodes peuvent être facilement transposées à l'optimisation multiobjectif. Cet ouvrage présente aussi les difficultés que l'on peut rencontrer lorsque l'on essaie de mesurer les performances d'un système.

[Hooker 95] Cet article critique les tests actuellement appliqués pour évaluer les performances des méthodes d'optimisation. Il propose une nouvelle voie pour l'évaluation des performances des méthodes d'optimisation.

Chapitre 8

Les fonctions de test des méthodes d'optimisation multiobjectif

8.1 Introduction

Nous avons vu que le domaine de l'optimisation multiobjectif est foisonnant. Les méthodes que l'on a à notre disposition sont nombreuses.

Cependant, certaines méthodes ne sont efficaces que sur des problèmes qui respectent certaines contraintes. Or, dans certains cas, il est très difficile, voire impossible, de dire si un problème d'optimisation multiobjectif respecte une hypothèse ou non.

C'est pour pouvoir classer les différentes méthodes de traitement d'un problème d'optimisation multiobjectif que l'on a introduit des problèmes tests. Chaque problème permet de tester une difficulté bien particulière (surface de compromis discontinue, non convexe, etc.).

8.2 Les problèmes tests de Deb

Avant de passer en revue les fonctions tests de l'optimisation multiobjectif, il faut se rappeler qu'une représentation de la surface de compromis est satisfaisante si la répartition des points sur celle-ci est uniforme.

Dans ce cas, si l'on n'est pas satisfait par une solution, on pourra en choisir une autre, située dans son voisinage. Du fait de l'uniformité de la représentation des solutions, la variation de valeur des fonctions objectif ne sera pas brusque lorsque l'on passe d'une solution à sa voisine (voir la figure 1.24).

Une fonction test est donc une fonction destinée à mettre en difficulté la méthode d'optimisation multiobjectif, dans la recherche d'une répartition uniforme des points solutions sur la surface de compromis.

On distingue deux familles de difficultés :

- celles qui empêchent la méthode de converger vers la surface de compromis ;
- celles qui empêchent d'aboutir à une répartition uniforme des solutions sur la surface de compromis.

Dans la première famille, on trouve la multifrontalité de la surface de compromis.

Dans la seconde famille, on trouve les difficultés suivantes :

- la non convexité,
- la discontinuité,

– la non uniformité de la surface de compromis.

Deb [Deb 98a] a proposé une série de fonctions de test basée sur le problème biobjectif “générique” suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) = f\left(x_1, \cdots, x_m\right) \\ \text{minimiser} & f_2\left(\overrightarrow{x}\right) = g\left(x_{m+1}, \cdots, x_N\right) \cdot h\left(f\left(x_1, \cdots, x_m\right), g\left(x_{m+1}, \cdots, x_N\right)\right) \\ \text{avec} & \overrightarrow{g}\left(\overrightarrow{x}\right) \leq 0 \\ \text{et} & \overrightarrow{h}\left(\overrightarrow{x}\right) = 0 \end{array}\right.$$

Dans ce problème générique, chaque fonction joue un rôle particulier :

f : contrôle l’uniformité de la répartition des solutions le long de la surface de compromis.

g : contrôle la “multifrontalité” de la surface de compromis. Elle permet aussi de créer un optimum isolé.

h : permet de contrôler l’aspect de la surface de compromis (convexe, non convexe, etc.).

8.2.1 La fonction non convexe de Deb

Comme on l’a vu avec la méthode de pondération des fonctions objectif, certaines méthodes réagissent très mal face à une non convexité. La plupart du temps, ces méthodes n’arrivent pas à couvrir une zone non convexe.

La fonction non convexe de Deb suivante permet de tester cet aspect :

$$\begin{array}{l} f\left(x_1\right) = 4 \cdot x_1 \\ g\left(x_2\right) = \begin{cases} 4 - 3 \cdot \exp\left(-\left(\frac{x_2 - 0.2}{0.02}\right)^2\right) & \text{si } 0 \leq x_2 \leq 0.4 \\ 4 - 2 \cdot \exp\left(-\left(\frac{x_2 - 0.7}{0.2}\right)^2\right) & \text{si } 0.4 < x_2 \leq 1 \end{cases} \\ h(f, g) = \begin{cases} 1 - \left(\frac{f}{\ \cdot g}\right) & \text{si } f \leq \ \cdot g \\ 0 & \text{sinon} \end{cases} \\ \quad = 0.25 + 3.75 \cdot \left(g\left(x_2\right) - 1\right) \text{ et } \quad = 1. \end{array}$$

Le problème est alors le suivant :

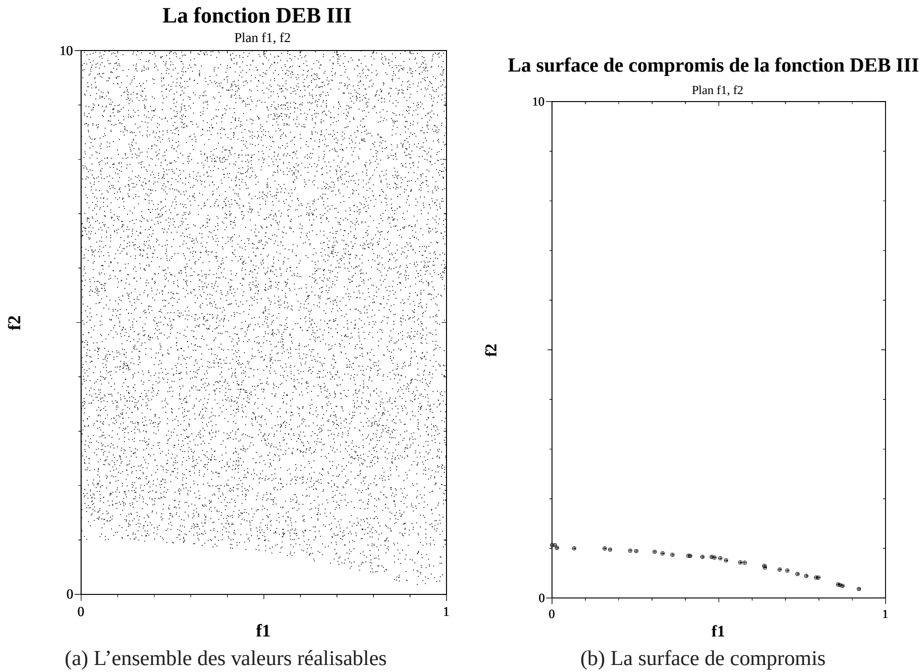
$$\left|\begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) = f\left(x_1\right) \\ \text{minimiser} & f_2\left(\overrightarrow{x}\right) = g\left(x_2\right) \cdot h(f, g) \\ \text{avec} & x_1, x_2 \in [0, 1] \end{array}\right.$$

L’ensemble des valeurs réalisables et la surface de compromis sont représentés à la figure 8.1.

Pour obtenir ce type de représentation de l’ensemble solution d’un problème test, on a appliqué l’algorithme 8.1.

Un avantage de ce type de problème est que l’on peut obtenir l’expression analytique de la surface de compromis. En effet, celle-ci se situe en $(x_1, 0)$, avec $x_1 \in [0, 1]$.

Pour ce problème, l’expression analytique de la surface de compromis est :



(a) L'ensemble des valeurs réalisables

(b) La surface de compromis

FIG. 8.1 – L'allure de la fonction non convexe de Deb.

Algorithm 8.1 Calcul aléatoire de l'allure d'un ensemble solution.

---

x_1, x_2 sont les variables de décision du problème.
$x_{1_{max}}$, $x_{1_{min}}$, $x_{2_{max}}$, $x_{2_{min}}$ sont les bornes de l'espace de décision
Const $\left(\overrightarrow{f}(x_1, x_2)\right)$ est une fonction qui vaut 1 si les contraintes du problème sont respectées et 0 sinon
N est le nombre de points à calculer pour la représentation de l'ensemble solution.
$I = 0$ est un index.
rand (A, B) est une fonction qui retourne un nombre au hasard, compris entre A et B.

while($I < N$)
  $x_1 = \text{rand}(x_{1_{min}}, x_{1_{max}})$
  $x_2 = \text{rand}(x_{2_{min}}, x_{2_{max}})$
  if Const $\left(\overrightarrow{f}(x_1, x_2)\right) = 1$ afficher $x_1, x_2, \overrightarrow{f}(x_1, x_2)$ et $I = I + 1$
end while

$$\left| \begin{array}{l} f(x_1) = 4 \cdot x_1 \\ g(x_1) = 4 - 3 \cdot \exp(-100) \\ h(x_1) = 1 - \left(\frac{4 \cdot x_1}{4 - 3 \cdot \exp(-100)}\right) \quad \text{si } f(x_1) \leq g(x_1) \\ \text{avec } x_1 \in [0,1] \text{ et } \quad = 0.25 + 3.75 \cdot (3 \cdot (1 - \exp(-100))) \end{array} \right.$$

donc,

$$\left| \begin{array}{l} f_1(x_1) = 4 \cdot x_1 \\ f_2(x_1) = \begin{cases} (4 - 3 \cdot \exp(-100)) \cdot \left(1 - \left(\frac{4 \cdot x_1}{4 - 3 \cdot \exp(-100)}\right) \right) & \text{si } f(x_1) \leq g(x_1) \\ 0 & \text{sinon} \end{cases} \\ \text{avec } x_1 \in [0,1] \text{ et } \quad = 0.25 + 3.75 \cdot (3 \cdot (1 - \exp(-100))) \end{array} \right.$$

si $\exp(-100) \simeq 0$ alors,

$$\left| \begin{array}{l} f_1(x_1) = 4 \cdot x_1 \\ f_2(x_1) = \begin{cases} 4 \cdot \left(1 - x_1^{11.5}\right) & \text{si } f(x_1) \leq g(x_1) \\ 0 & \text{sinon} \end{cases} \\ \text{avec } x_1 \in [0,1] \end{array} \right.$$

8.2.2 La fonction discontinue de Deb

La grande majorité des méthodes d'optimisation multiobjectif ne parviennent pas à assurer l'uniformité de la répartition des solutions sur la surface de compromis lorsqu'elles sont confrontées à une discontinuité.

La fonction discontinue de Deb est définie comme suit :

$$\begin{array}{l} f(x_1) = x_1 \\ g(x_2) = 1 + 10 \cdot x_2 \\ h(f,g) = 1 - \left(\frac{f}{g}\right) \quad - \left(\frac{f}{g}\right) \cdot \sin(2\Pi \cdot q \cdot f) \end{array}$$

Le paramètre q permet de définir le nombre de régions discontinues dans l'intervalle $[0,1]$. Le plus souvent, le paramètre vaut 2.

Le problème est alors défini comme suit :

$$\left| \begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) = f(x_1) \\ \text{minimiser} & f_2(\overrightarrow{x}) = g(x_2) \cdot h(f,g) \\ \text{avec} & x_1, x_2 \in [0,1] \end{array} \right.$$

L'ensemble des valeurs réalisables et la surface de compromis sont représentés à la figure 8.2.

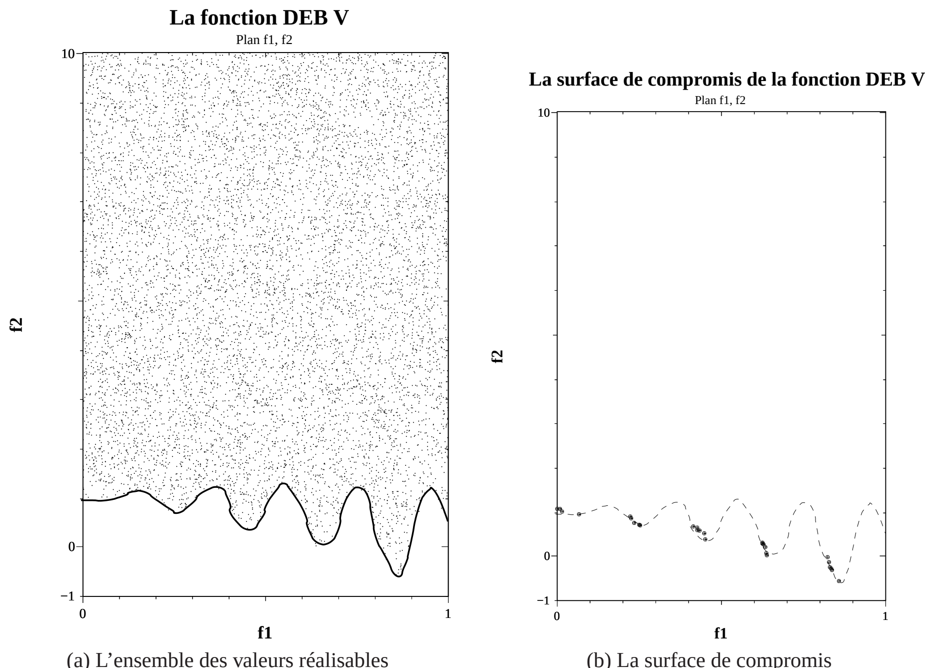
L'expression analytique de l'enveloppe de la surface de compromis est :

$$\left| \begin{array}{l} f_1(x_1) = x_1 \\ f_2(x_1) = 1 - x_1^2 - x_1 \cdot \sin(2\Pi \cdot q \cdot x_1) \\ \text{avec } x_1 \in [0,1] \end{array} \right.$$

Donc, on a :

$$f_2 = 1 - f_1^2 - f_1 \cdot \sin(2\Pi \cdot q \cdot f_1) \tag{8.1}$$

Il ne faut pas oublier d'éliminer les points de l'enveloppe qui n'appartiennent pas à la surface de compromis.



(a) L'ensemble des valeurs réalisables

(b) La surface de compromis

FIG. 8.2 – L'allure de la fonction discontinue de Deb.

8.2.3 La fonction multifrontale de Deb

Une autre grande difficulté que l'on peut rencontrer est la multifrontalité. On la rencontre lorsqu'un problème d'optimisation multiobjectif possède une ou plusieurs surfaces de compromis “fantômes”.

Une surface de compromis fantôme est, le plus souvent, due à l'existence d'un optimum global isolé et d'un optimum local. Lors du processus d'optimisation, les solutions convergent vers l'optimum local et forment ainsi une surface de compromis, qui est différente de la surface de compromis “optimale”.

La fonction multifrontale de Deb est définie comme suit :

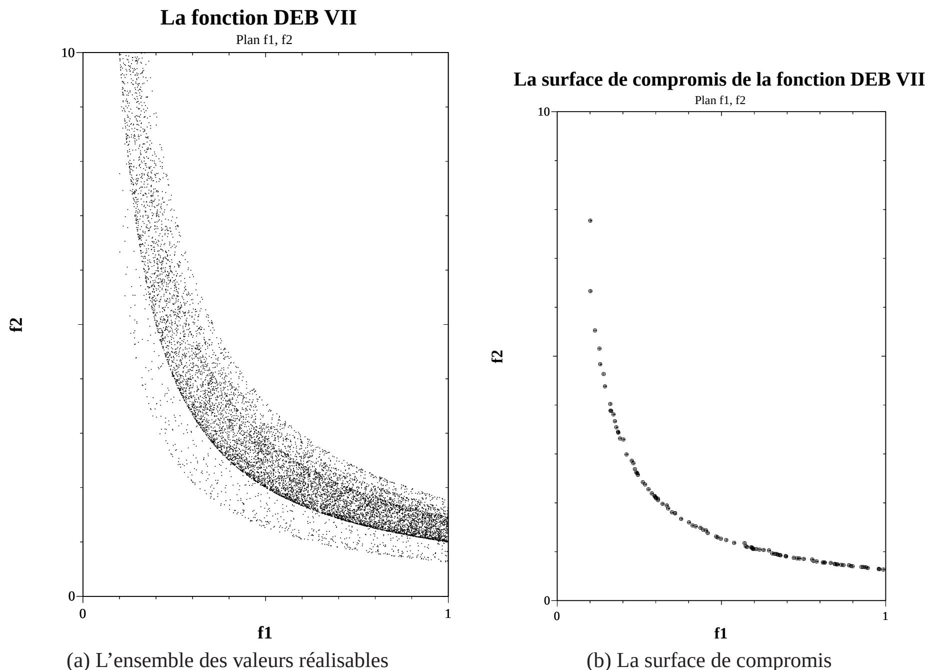
$$f(x_1) = x_1$$
$$g(x_2) = 2 - \exp\left(-\left(\frac{x_2-0.2}{0.04}\right)^2\right) - 0.8 \cdot \exp\left(-\left(\frac{x_2-0.6}{0.4}\right)^2\right)$$
$$h(f,g) = \frac{1}{f}$$

Le problème est alors défini comme suit :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) = f(x_1) \\ \text{minimiser} & f_2(\overrightarrow{x}) = g(x_2) \cdot h(f,g) \\ \text{avec} & x_1, x_2 \in [0,1] \end{array}\right.$$

L'ensemble des valeurs réalisables et la surface de compromis sont représentés à la figure 8.3.

L'expression analytique de la surface de compromis est la suivante :



(a) L'ensemble des valeurs réalisables

(b) La surface de compromis

FIG. 8.3 – L'allure de la fonction multifrontale de Deb.

$f_1(x_1) = x_1$
$f_2(x_1) = \left(2 - \exp(-25) - 0.8 \cdot \exp\left(\frac{3}{2}\right)\right) \cdot \frac{1}{x_1}$
avec $x_1 \in [0,1]$

On a donc :

$$f_2 = \frac{1.9156806}{f_1} \tag{8.2}$$

8.2.4 La fonction non uniforme de Deb

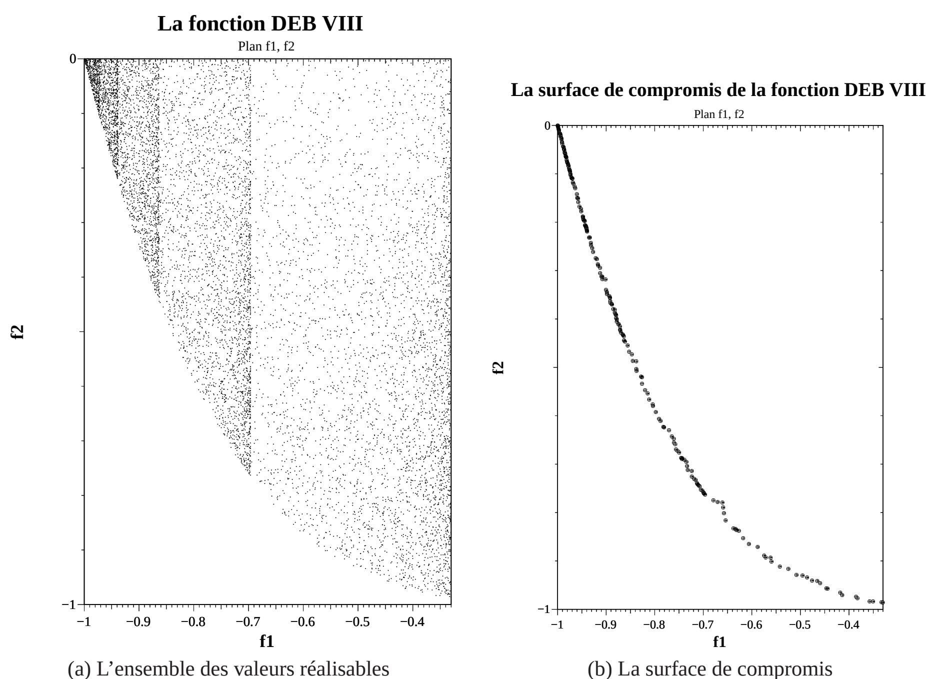
Dans certains cas, du fait de la structure des fonctions objectif à optimiser, les solutions vont se concentrer autour de certains points. Dans ce cas, on a un problème de non-uniformité de la représentation des solutions sur la surface de compromis.

La fonction test non uniforme de Deb est définie comme suit :

$f(x_1) = 1 - \exp(-4 \cdot x_1) \cdot \sin^4(5\Pi \cdot x_1)$
$g(x_2) = 1 - 10 \cdot x_2$
$h(f,g) = \sqrt{1-f}$

minimiser	$f_1(\overrightarrow{x}) = -f(x_1)$
minimiser	$f_2(\overrightarrow{x}) = -g(x_2) \cdot h(f,g)$
avec	$x_1, x_2 \in [0,1]$

L'ensemble des valeurs réalisables et la surface de compromis sont représentés à la figure 8.4.



(a) L'ensemble des valeurs réalisables

(b) La surface de compromis

FIG. 8.4 – L'allure de la fonction non uniforme de Deb.

L'expression analytique de la surface de compromis est la suivante :

$$f_2 = -\sqrt{1 - f_1} \tag{8.3}$$

avec $f_1 \in [-1, 0]$.

8.3 Les problèmes tests de Hanne

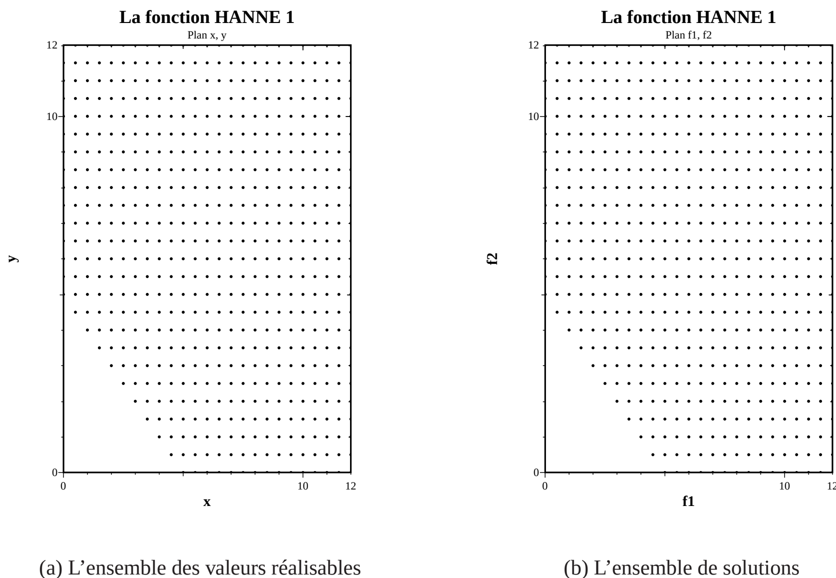
La principale différence avec les problèmes tests de Deb réside dans l'introduction d'une contrainte. Pour le reste, les problèmes tests de Hanne couvrent à peu près le même champ que les problèmes tests de Deb.

8.3.1 La fonction linéaire de Hanne

Ce problème test est le suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) = x_1 \\ \text{minimiser} & f_2(\overrightarrow{x}) = x_2 \\ \text{avec} & x_1 \geq 0 \\ & x_2 \geq 0 \\ \text{et} & x_1 + x_2 \geq 5 \end{array}\right.$$

L'ensemble des valeurs réalisables et la surface de compromis sont représentés sur les figures 8.5 et 8.6.



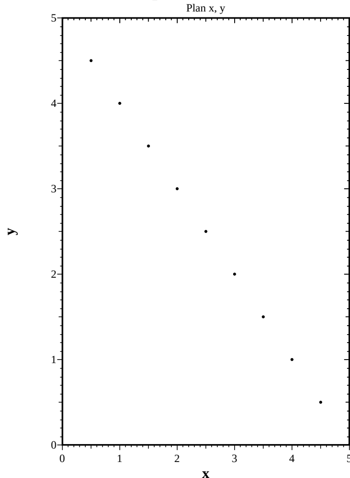
(a) L'ensemble des valeurs réalisables

(b) L'ensemble de solutions

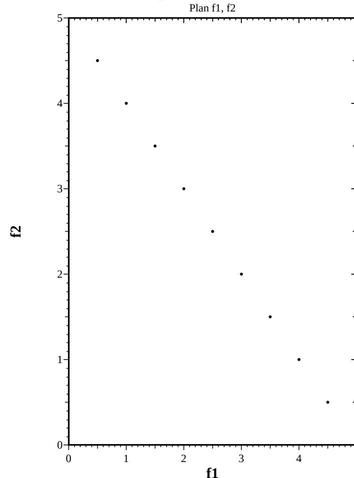
FIG. 8.5 – L'allure de la fonction linéaire de Hanne (ensemble de solutions).

A la figure 8.5-a, on a représenté l'ensemble de solutions dans l'espace de la variable de décision $\overrightarrow{x}$. A la figure 8.5-b, on a représenté l'ensemble de solutions dans l'espace des fonctions objectif. On obtient ces deux ensembles en appliquant l'algorithme 8.2.

A la figure 8.6-a, on a représenté la surface de compromis du problème test dans l'espace des variables de décision. A la figure 8.6-b on a représenté la surface de compromis dans l'espace des fonctions objectif. Ce type de représentation (espace de décision - espace des fonctions objectif) nous permet de visualiser des difficultés possibles dans l'espace de décision (l'ensemble solution peut être morcelé dans l'espace des variables de décision par exemple).



(a) L'ensemble des valeurs de *x* associées à la surface de compromis



(b) La surface de compromis

FIG. 8.6 – L'allure de la fonction linéaire de Hanne (surface de compromis).

Algorithm 8.2 Calcul séquentiel de l'allure d'un ensemble solution.

x_1, x_2 variables de décision du problème test

$x_{1_{max}}, x_{1_{min}}, x_{2_{max}}, x_{2_{min}}$ bornes des variables de décision.

Δx_1, Δx_2 pas de variation des variables de décision

Const $\left(\overrightarrow{f}(x_1, x_2)\right)$ fonction qui vaut 1 si les contraintes du problème considéré sont vérifiées et 0 sinon.

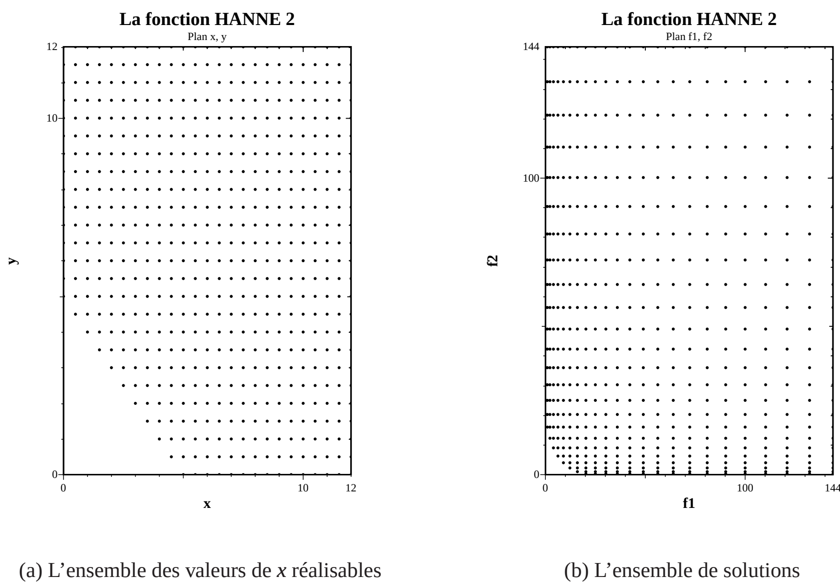
$x_1 = x_{1_{min}}$
$x_2 = x_{2_{min}}$
do
  do
    si Const $\left(\overrightarrow{f}(x_1, x_2)\right) = 1$ afficher x_1, x_2, $\overrightarrow{f}(x_1, x_2)$
    $x_1 = x_1 + \Delta x_1$
  while $(x_1 < x_{1_{max}})$
  $x_2 = x_2 + \Delta x_2$
while $(x_2 < x_{2_{max}})$

8.3.2 La fonction convexe de Hanne

Ce problème test est le suivant :

minimiser	$f_1(\overrightarrow{x}) = x_1^2$
minimiser	$f_2(\overrightarrow{x}) = x_2^2$
avec	$x_1 \geq 0$
	$x_2 \geq 0$
et	$x_1 + x_2 \geq 5$

L'ensemble des valeurs réalisables et la surface de compromis sont représentés sur les figures 8.7 et 8.8.

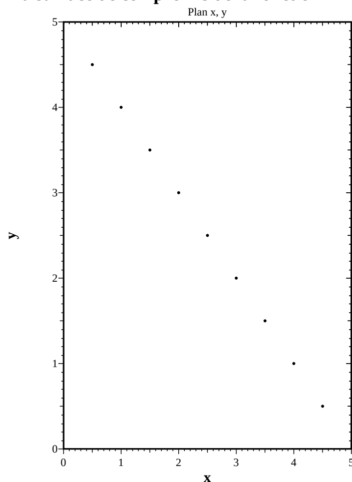


(a) L'ensemble des valeurs de x réalisables (b) L'ensemble de solutions

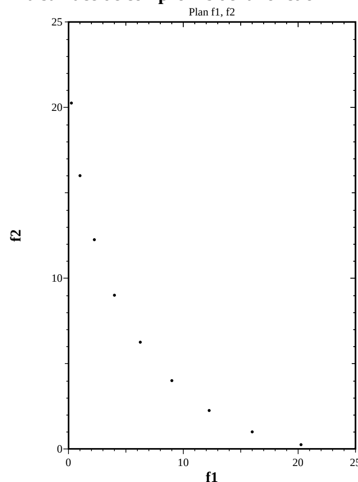
FIG. 8.7 – L'allure de la fonction convexe de Hanne (ensemble de solutions).

A la figure 8.7-a et 8.7-b, on a représenté l'allure de l'ensemble de solutions respectivement dans l'espace de la variable de décision et dans l'espace des fonctions objectif. Ici, contrairement au problème précédent (problème linéaire de Hanne), les deux allures ne se ressemblent pas, car l'ensemble solution dans l'espace des fonctions objectif n'est plus une simple copie de l'ensemble de solutions dans l'espace des variables de décision.

Un avantage de la méthode de calcul séquentiel de l'allure de l'ensemble de solutions est, comme on peut le voir en comparant l'espace des variables de décision et l'espace des fonctions objectif, que l'on peut visualiser l'effet des fonctions objectif sur la déformation de l'ensemble de solutions.



(a) L'ensemble des valeurs de x associées à la surface de compromis



(b) La surface de compromis

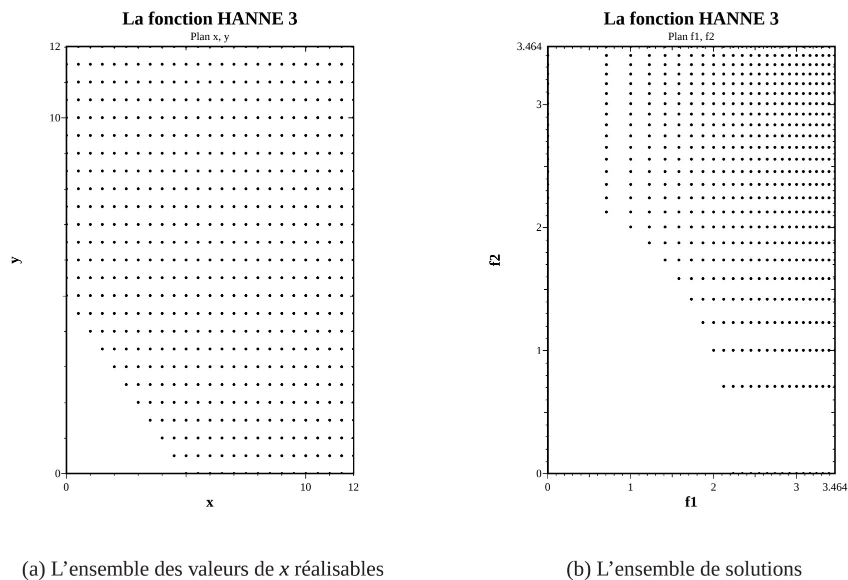
FIG. 8.8 – L'allure de la fonction convexe de Hanne (surface de compromis).

8.3.3 La fonction non convexe de Hanne

Ce problème test est le suivant :

$$
\left|
\begin{array}{ll}
\text{minimiser} & f_1\left(\overrightarrow{x}\right) = \sqrt{x_1} \\
\text{minimiser} & f_2\left(\overrightarrow{x}\right) = \sqrt{x_2} \\
\text{avec} & x_1 \geq 0 \\
 & x_2 \geq 0 \\
\text{et} & x_1 + x_2 \geq 5
\end{array}
\right.
$$

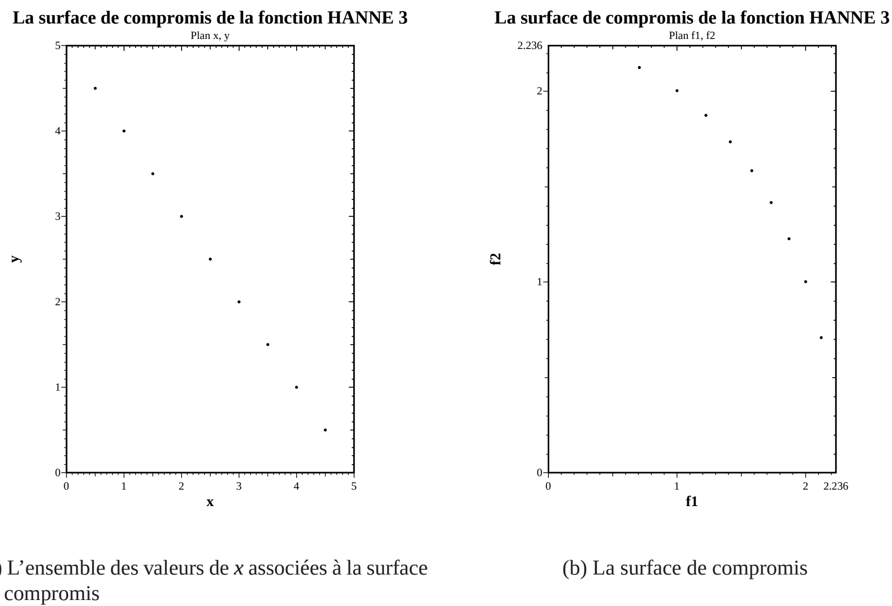
L'ensemble des valeurs réalisables et la surface de compromis sont représentés sur les figures 8.9 et 8.10.



(a) L'ensemble des valeurs de *x* réalisables

(b) L'ensemble de solutions

FIG. 8.9 – L'allure de la fonction non convexe de Hanne (ensemble de solutions).



(a) L'ensemble des valeurs de *x* associées à la surface de compromis

(b) La surface de compromis

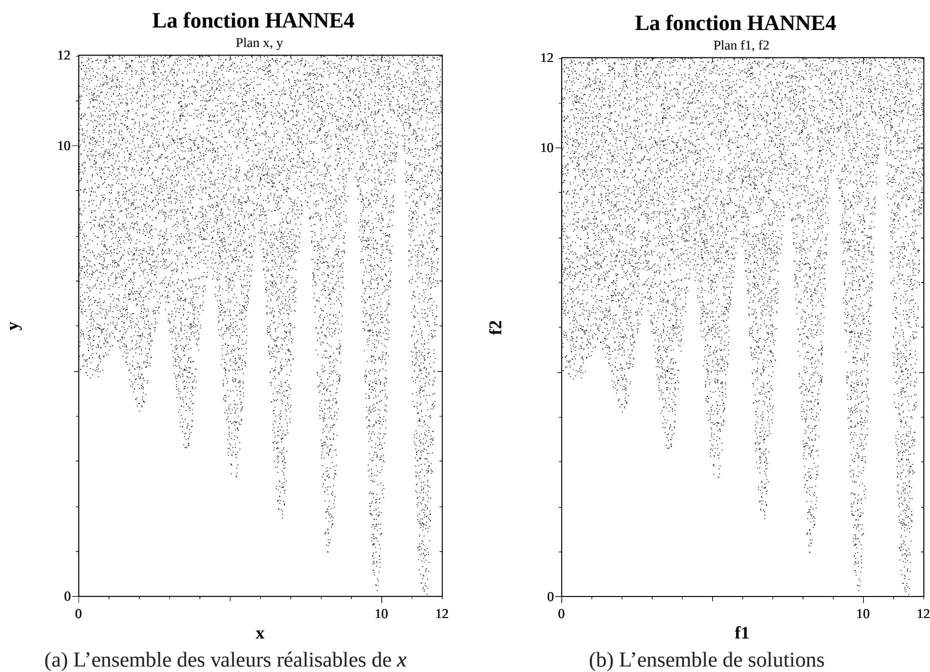
FIG. 8.10 – L'allure de la fonction non convexe de Hanne (surface de compromis).

8.3.4 La fonction discontinue de Hanne

Ce problème test est le suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) = x_1 \\ \text{minimiser} & f_2\left(\overrightarrow{x}\right) = x_2 \\ \text{avec} & x_1 \geq 0 \\ & x_2 \geq 0 \\ \text{et} & x_2 - 5 + 0.5 \cdot x_1 \cdot \sin\left(4 \cdot x_1\right) \geq 0 \end{array}\right.$$

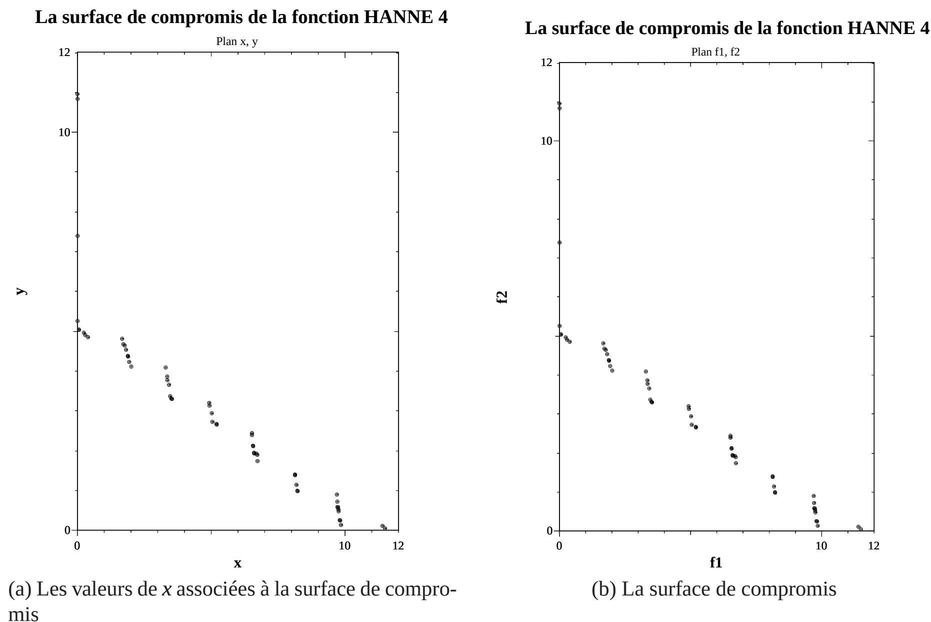
L'ensemble des valeurs réalisables et la surface de compromis sont représentés sur les figures 8.11 et 8.12.



(a) L'ensemble des valeurs réalisables de x

(b) L'ensemble de solutions

FIG. 8.11 – L'allure de la fonction discontinue de Hanne (ensemble solution).



(a) Les valeurs de x associées à la surface de compromis

(b) La surface de compromis

FIG. 8.12 – L'allure de la fonction discontinue de Hanne (surface de compromis).

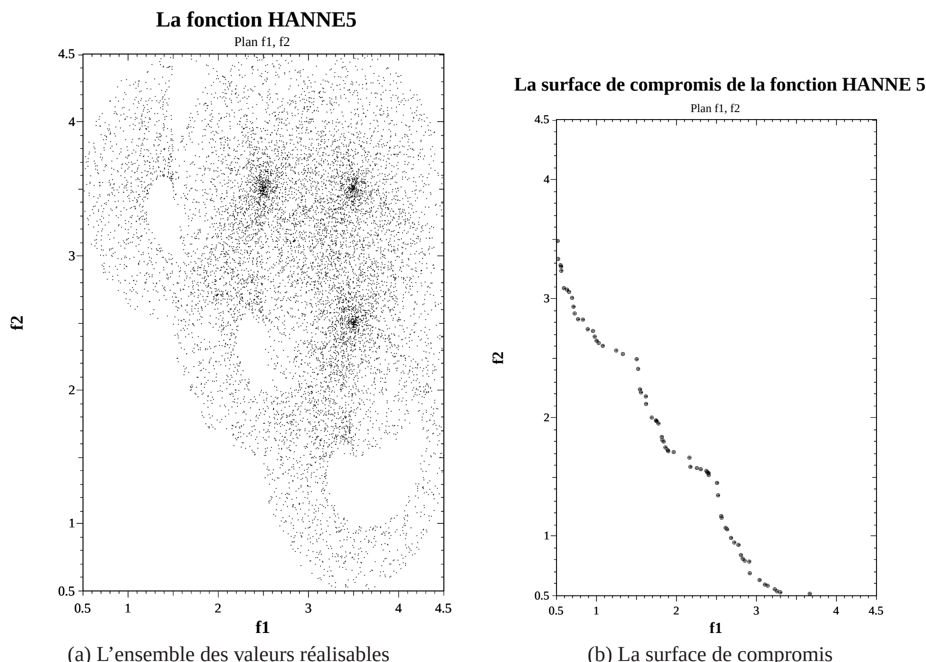
8.3.5 La fonction à plusieurs zones efficaces de Hanne

Ce problème test est le suivant :

$$\left|\begin{array}{ll} \text{minimiser} & f_1\left(\overrightarrow{x}\right) = int\left(x_1\right) + 0.5 + \left(x_1 - int\left(x_1\right)\right) \cdot \sin\left(2\Pi \cdot \left(x_2 - int\left(x_2\right)\right)\right) \\ \text{minimiser} & f_2\left(\overrightarrow{x}\right) = int\left(x_2\right) + 0.5 + \left(x_1 - int\left(x_1\right)\right) \cdot \cos\left(2\Pi \cdot \left(x_2 - int\left(x_2\right)\right)\right) \\ \text{avec} & x_1 \geq 0 \\ & x_2 \geq 0 \\ \text{et} & x_1 + x_2 \geq 5 \end{array}\right.$$

où $int\left(x\right)$ désigne la partie entière de x.

L'ensemble des valeurs réalisables et la surface de compromis sont représentés sur la figure 8.13.



(a) L'ensemble des valeurs réalisables

(b) La surface de compromis

FIG. 8.13 – L'allure de la fonction à plusieurs zones efficaces de Hanne.

8.4 Bibliographie commentée

[Deb 98a] Cet article présente une méthode pour construire des problèmes tests. Cette expression générique permet de construire tous les types de problèmes tests : convexe, non convexe, avec optima locaux, etc. Ces problèmes tests sont très utilisés lors de comparaisons entre méthodes.

[Whitley et al. 96] Cet article traite des problèmes tests monobjectif. L'auteur souligne un certain nombre d'incohérences avec certains problèmes tests actuels. Il propose ensuite des règles que devraient respecter tous les problèmes tests.

Chapitre 9

Tentatives de classification des méthodes d'optimisation multiobjectif

9.1 Introduction

Comme on a pu le voir tout au long de cet ouvrage, il existe un très grand nombre de méthodes d'optimisation multiobjectif. Cette situation pose problème lorsqu'il faut choisir une de ces méthodes d'optimisation pour résoudre un problème concret. Aujourd'hui, peu de travaux existent sur la classification des méthodes d'optimisation.

Cependant, on trouvera dans [Miettinen 99] un organigramme permettant de choisir une méthode d'optimisation multiobjectif en fonction du type de problème que l'on souhaite résoudre. Cet organigramme est très compliqué et, comme on le verra par la suite, l'approche proposée n'est peut-être pas la meilleure pour ce genre de situation. Par ailleurs, on trouvera dans [Talbi 98] une technique permettant de classer les métaheuristiques hybrides.

Dans ce chapitre, nous présentons deux approches d'une telle classification :

- une approche plus mathématique, qui s'efforce de décrire le plus succinctement possible une méthode d'optimisation ;
- une approche graphique, où l'on cherche une décomposition hiérarchique des méthodes d'optimisation.

Dans tout le chapitre, nous faisons la distinction entre les termes suivants :

- méthode de traitement multiobjectif ;
- méthode d'optimisation ;
- méthode d'optimisation multiobjectif.

Nous désignons par "méthode de traitement multiobjectif" la méthode utilisée pour transformer le problème multiobjectif original en un problème multiobjectif adapté à la résolution. Cette méthode peut être, par exemple, la méthode de pondération des fonctions objectif.

Nous désignons ensuite par "méthode d'optimisation" la méthode de recherche utilisée pour résoudre un problème multiobjectif. La méthode de recherche n'inclut pas la méthode de transformation du problème multiobjectif.

Pour finir, nous désignons par "méthode d'optimisation multiobjectif" un couple méthode de traitement du problème multiobjectif + méthode d'optimisation.

Nous allons maintenant étudier l'approche mathématique de la classification, puis nous nous intéresserons à l'approche graphique. Nous montrerons comment on obtient une dé-

composition hiérarchique de la structure d'une méthode d'optimisation, mais surtout, nous découvrirons qu'une classification des méthodes d'optimisation multiobjectif doit être vue comme un outil d'aide au questionnement, en vue de choisir une méthode, plutôt qu'un outil de sélection d'une méthode.

Nous illustrerons chaque type de classification des méthodes d'optimisation multiobjectif (mathématique et graphique) par quelques exemples.

9.2 Classification "mathématique" des méthodes d'optimisation

9.2.1 Introduction

Cette approche ne considère que la manière de transformer un problème d'optimisation multiobjectif en un autre problème d'optimisation multiobjectif (voir [Ehrgott 00]). Elle ne s'occupe pas de décrire les méthodes d'optimisation en elles-mêmes.

9.2.2 La formule de classification de Erghott

Considérons le problème d'optimisation multiobjectif suivant :

$$\min_{\overrightarrow{x} \in X} \overrightarrow{f}\left(\overrightarrow{x}\right)$$

Ici, l'expression *min* devrait être écrite entre guillemets car, pour les problèmes multiobjectifs, tant que la relation d'ordre n'a pas été spécifiée, le sens de l'expression *min* n'est pas suffisamment précis pour que l'optimisation puisse être effectuée.

Si maintenant on précise que la relation d'ordre que l'on va considérer pour ce problème est la dominance au sens de Pareto, alors le sens de l'expression *min* est précisé : nous cherchons le minimum, au sens de Pareto, sur l'espace des variables de décision X, du vecteur de fonctions objectif $\overrightarrow{f}\left(\overrightarrow{x}\right)$.

Les éléments nécessaires à la bonne spécification d'un problème d'optimisation multiobjectif sont les suivants :
- l'espace des variables de décision (X dans notre exemple) ;
- l'espace des fonctions objectif ($\mathbb{R}^N$ dans notre exemple) ;
- un espace ordonné ($\mathbb{R}^P$ avec la relation de dominance comme relation d'ordre, que nous noterons $\preceq$) ;
- un modèle de transformation (nous noterons ce modèle dans notre exemple). Nous préciserons par la suite la signification de ce terme.

M. Erghott [Ehrgott 00] se propose de référencer une méthode d'optimisation en utilisant l'expression suivante :

$$\left(X, \overrightarrow{f}, \mathbb{R}^N\right) / \quad / \left(\mathbb{R}^P, \preceq\right)$$

Cette expression reflète bien le processus de transformation d'un problème d'optimisation multiobjectif en un problème d'optimisation multiobjectif qui soit adapté à la résolution par une méthode de recherche d'optimum. Cette transformation est représentée à la figure 9.1.

Comme on le voit, nous distinguons ici la méthode de transformation d'un problème d'optimisation multiobjectif (la méthode de pondération des fonctions objectif est une méthode de transformation) et la méthode de recherche d'un optimum (le recuit simulé est une méthode de recherche d'optimum).

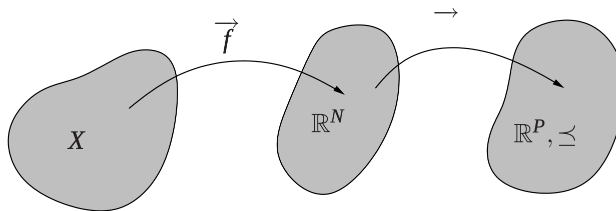


FIG. 9.1 – La transformation d’un problème d’optimisation multiobjectif.

Pour mieux comprendre l’expression de la classification, développons quelques exemples.

Exemple 1 : Se ramener à un problème de minimisation

Tous les problèmes d’optimisation ne se formulent pas sous la forme d’un problème de minimisation. Par exemple, pour le problème suivant, on a deux maximisations et une minimisation, ce qui induit la transformation indiquée :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{maximiser} & f_2(\overrightarrow{x}) \\ \text{maximiser} & f_3(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{x} \in X \subset \mathbb{R}^4 \end{array}\right. \quad \rightarrow \quad \left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{minimiser} & -f_2(\overrightarrow{x}) \\ \text{minimiser} & -f_3(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{x} \in X \subset \mathbb{R}^4 \end{array}\right.$$

Pour cette transformation du problème d’optimisation multiobjectif, le modèle de la transformation est le suivant :

$$(z_1, z_2, z_3) = (z_1, -z_2, -z_3)$$

L’espace des variables de décision est X.

L’espace des fonctions objectif est $\mathbb{R}^3$.

L’espace des fonctions objectif après transformation est $\mathbb{R}^3$.

L’ordre que l’on choisit sur cet espace est la relation de dominance sur l’espace $\mathbb{R}^3$ que nous noterons $\preceq$.

Par conséquent, l’expression de la classification de cette méthode est la suivante :

$$\left(X, (f_1, f_2, f_3), \mathbb{R}^3\right) / (z_1, -z_2, -z_3) / \left(\mathbb{R}^3, \preceq\right)$$

Exemple 2 : La méthode de Tchebychev

La transformation effectuée sur le problème multiobjectif est la suivante :

$$\left|\begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{maximiser} & f_2(\overrightarrow{x}) \\ \text{maximiser} & f_3(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{x} \in X \end{array}\right. \quad \rightarrow \quad \left|\begin{array}{ll} \text{minimiser} & \max\left(f_1(\overrightarrow{x}), -f_2(\overrightarrow{x}), -f_3(\overrightarrow{x})\right) \\ \text{avec} & \overrightarrow{x} \in X \end{array}\right.$$

Pour cette transformation du problème d’optimisation multiobjectif, le modèle de la transformation est le suivant :

$$(z_1, z_2, z_3) = \max(z_1, -z_2, -z_3)$$

L’espace des variables de décision est X.

L’espace des fonctions objectif est $\mathbb{R}^3$.

L'espace des fonctions objectif après transformation est $\mathbb{R}$.
L'ordre que l'on choisit sur cet espace est la relation d'ordre classique : $\leq$.
L'expression de la classification de la méthode de Tchebychev est la suivante :

$$\left(X, (f_1, f_2, f_3), \mathbb{R}^3\right) / \max / (\mathbb{R}, \leq)$$

Exemple 3 : La méthode de pondération des fonctions objectif

La transformation effectuée sur le problème multiobjectif est la suivante :

$$\left| \begin{array}{ll} \text{minimiser} & f_1(\overrightarrow{x}) \\ \text{maximiser} & f_2(\overrightarrow{x}) \\ \text{maximiser} & f_3(\overrightarrow{x}) \\ \text{avec} & \overrightarrow{x} \in X \end{array} \right. \rightarrow \left| \begin{array}{ll} \text{minimiser} & \begin{array}{l} {}_1 \cdot f_1(\overrightarrow{x}) - {}_2 \cdot f_2(\overrightarrow{x}) \\ - {}_3 \cdot f_3(\overrightarrow{x}) \end{array} \\ \text{avec} & \overrightarrow{x} \in X \\ \text{et} & {}_i \geq 0 \end{array} \right.$$

Pour cette transformation du problème d'optimisation multiobjectif, le modèle de la transformation est le suivant :

$$(z_1, z_2, z_3) = {}_1 \cdot z_1 - {}_2 \cdot z_2 - {}_3 \cdot z_3$$

L'espace des variables de décision est X.
L'espace des fonctions objectif est $\mathbb{R}^3$.
L'espace des fonctions objectif après transformation est $\mathbb{R}$.
L'ordre que l'on choisit sur cet espace est la relation d'ordre classique : $\leq$.
L'expression de la classification de la méthode de Tchebychev est la suivante :

$$\left(X, (f_1, f_2, f_3), \mathbb{R}^3\right) / {}_1 \cdot z_1 - {}_2 \cdot z_2 - {}_3 \cdot z_3 / (\mathbb{R}, \leq)$$

9.2.3 Conclusion

Cette classification présente le grand avantage d'être applicable aussi bien à des problèmes d'optimisation qu'à des méthodes d'optimisation (voir [Ehrgott et al. 00]).

Cette faculté de description commune permet à l'utilisateur de comparer la description du problème avec la description de la méthode de résolution et ainsi de sélectionner des méthodes qui ont certains points communs avec la description du problème, car elles sont le plus susceptibles d'être adaptées à la résolution de ce problème.

9.3 La classification hiérarchique des méthodes d'optimisation multiobjectif

9.3.1 Introduction

Cette approche peut présenter certains inconvénients comme, par exemple, une explosion du nombre de niveaux hiérarchiques due au nombre important des éléments distinctifs décrivant une méthode d'optimisation. Il est cependant possible de limiter cet inconvénient en choisissant soigneusement les éléments descriptifs d'une méthode d'optimisation et en limitant le domaine d'intervention de la classification.

9.3.2 Une hiérarchie graphique

9.3.2.1 Une hiérarchie pour le traitement du problème multiobjectif

Nous avons déjà rencontré certains éléments de description des méthodes d'optimisation multiobjectif au cours de la présentation des différentes méthodes.

Par exemple, en optimisation multiobjectif, on peut distinguer trois approches différentes suivant la connaissance que l'on a du problème à résoudre. On peut, en effet, résoudre ce problème :

– soit *a priori* si l'on connaît suffisamment le problème et que l'on est capable de fournir avec précision ses préférences vis-à-vis des différentes fonctions objectif ;
– soit *a posteriori* si l'on ne connaît pas le problème. On cherche alors à approcher la surface de compromis en la couvrant avec un nombre fini de solutions ;
– soit *progressivement*, en dialoguant avec le décideur de manière à ce qu'il puisse préciser ses préférences.

Cette première étape dans la description des méthodes d'optimisation nous donne trois branches distinctes (voir figure 9.3).

La branche des méthodes progressives est bien déconnectée des deux autres (à cause, en particulier, de la phase d'interaction qui différencie les méthodes progressives des méthodes *a priori* et *a posteriori*).

Les branches des méthodes *a priori* et *a posteriori* ne sont pas indépendantes dans le sens où une méthode *a priori* peut très bien être utilisée comme une méthode *a posteriori* (on relance la méthode d'optimisation a priori sur plusieurs préférences du décideur). En revanche, la réciproque n'est pas nécessairement vraie : il n'est pas intéressant d'utiliser une méthode *a posteriori* comme une méthode *a priori* (engendrer une approximation de la surface de compromis et ne conserver que la solution qui représente le compromis le plus proche de celui formulé par le décideur).

Un autre degré que l'on pourra ajouter dans cette hiérarchie vient de la façon de traiter les fonctions objectif :

– On peut chercher à se ramener à une fonction monobjectif : c'est l'approche "agrégative". Cette approche peut ensuite se décomposer en trois autres niveaux :
  • l'agrégation par contraintes :
    la méthode de la contrainte est un exemple de ce type d'agrégation ;
  • l'agrégation sans contraintes :
    la méthode de pondération des fonctions objectif en est un exemple ;
  • l'agrégation hybride :
    la méthode de Corley en est un exemple.
– On peut traiter le problème multiobjectif de manière vectorielle, c'est-à-dire sans effectuer l'agrégation des fonctions objectif.
  La méthode MOGA est un exemple de ce type de traitement du problème multiobjectif.

Le dernier niveau, spécifiquement multiobjectif, que l'on peut considérer, concerne "l'imbrication" de la méthode de traitement multiobjectif dans la méthode d'optimisation :

– On peut avoir à utiliser des méthodes de traitement multiobjectif complètement découplées de la méthode d'optimisation. On parlera alors de méthode de traitement multiobjectif découplée.
  Par exemple, la méthode de pondération des fonctions objectif est une méthode de traitement du problème multiobjectif découplée, car elle ne nécessite pas de méthode d'optimisation dédiée pour la recherche de solutions.
– On peut aussi avoir à utiliser des méthodes de traitement multiobjectif qui ne laissent pas le choix quant à la méthode d'optimisation à utiliser. Par exemple, la méthode MOGA est une méthode d'optimisation multiobjectif dans laquelle le traitement du

problème multiobjectif est fortement couplé avec la méthode d'optimisation : on ne peut pas utiliser cette méthode avec autre chose qu'un algorithme génétique.

On obtient alors les hiérarchies des figures 9.2 et 9.3.

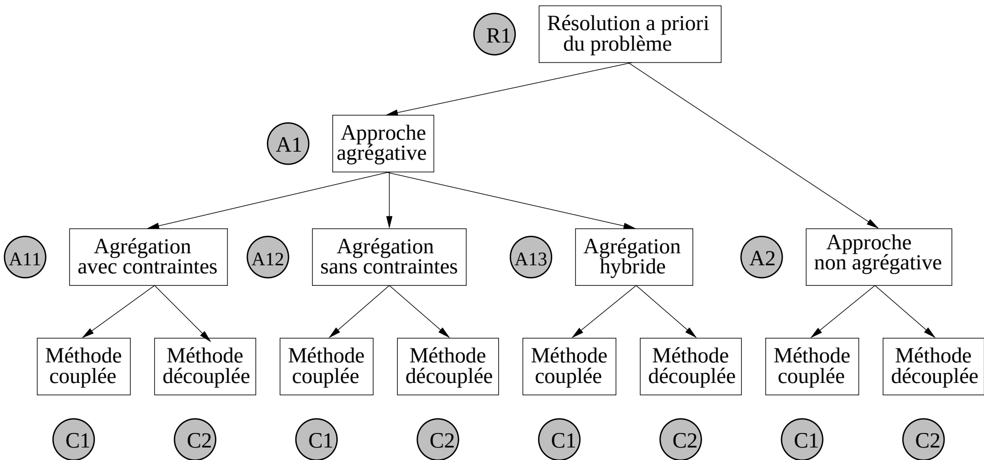


FIG. 9.2 – La hiérarchie des méthodes *a priori*.

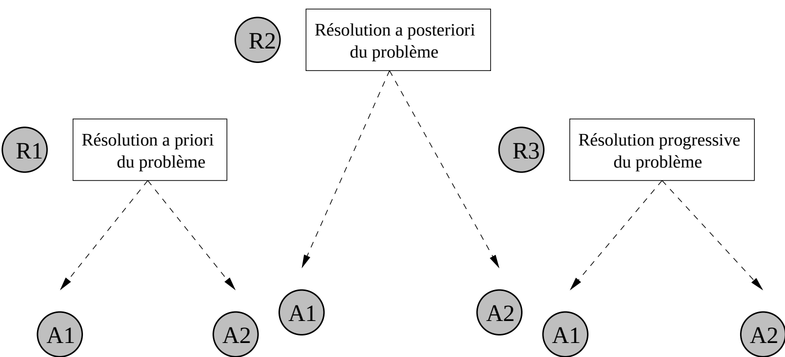


FIG. 9.3 – Le niveau supérieur hiérarchique des méthodes de transformation d'un problème multiobjectif.

La première constatation que l'on peut faire est la présence d'une branche non agrégative dans les racines R1 et R3. Ce n'est pas choquant pour la racine R3 (résolution progressive d'un problème d'optimisation multiobjectif), dans le sens où certaines méthodes (non présentées dans ce livre) proposent à l'utilisateur de sélectionner les individus qu'il préfère. En revanche, pour la racine R1 (résolution *a priori* d'un problème d'optimisation multiobjectif), la branche non agrégative illustre l'existence de méthodes du type ELECTRE (méthodes d'aide à la décision), dans lesquelles on fabrique une relation d'ordre à partir des préférences du décideur. La relation d'ordre ainsi construite peut être de forme agrégative ou de forme non agrégative. Dans ce dernier cas, la méthode classera les individus dans un ordre reflétant les préférences du décideur. Ces méthodes ne procurent donc pas un seul et unique individu, mais une famille complète de solutions.

Cherchons maintenant à traiter quelques exemples.

Exemple 1 : La méthode de pondération des fonctions objectif

Cette méthode est plutôt destinée à une résolution ***a priori*** du problème multiobjectif. Elle transforme le problème multiobjectif de manière à avoir à traiter une fonction monobjectif : c'est donc une approche agrégative sans contraintes. En outre, cette méthode est fortement découplée de la méthode d'optimisation, dans le sens où l'on peut utiliser n'importe quelle méthode d'optimisation pour trouver une solution à ce problème.

Le trajet dans la hiérarchie sera le suivant :

$$R1/A11/C2$$

La méthode de pondération des fonctions objectif peut être aussi utilisée dans des résolutions ***a posteriori*** ou *progressives*. Dans ce cas, la méthode est utilisée dans un système, et non sous sa forme de base.

Exemple 2 : La méthode MOGA

C'est une méthode de résolution ***a posteriori***. Elle engendre une famille de solutions pour un problème multiobjectif donné.

La méthode de traitement du problème multiobjectif est non agrégative. On ne demande pas à l'utilisateur de formuler des préférences. La méthode utilise directement la relation de dominance pour résoudre le problème.

La méthode de traitement du problème multiobjectif est fortement couplée avec une méthode d'optimisation, comme nous l'avons souligné plus haut.

Le trajet dans la hiérarchie sera le suivant :

$$R2/A2/C1$$

9.3.2.2 Une hiérarchie pour les interactions

Dans cette hiérarchie, des éléments manquent pour décrire les méthodes progressives. En effet, pour ces méthodes, le mode d'interaction est un élément discriminant.

Lorsque l'on analyse les modes de fonctionnement des méthodes interactives, on peut les classer en quatre catégories :

- les interactions via la formulation des préférences, nous noterons ce niveau de discrimination $I1$;
- les interactions via la formulation des contraintes, nous noterons ce niveau $I2$;
- les interactions hybrides via la formulation de préférences et de contraintes, nous noterons ce niveau $I3$;
- les interactions d'un autre type (comme, par exemple, par la sélection d'une sous-population de solutions intéressantes), nous noterons ce niveau $I4$.

Il nous faut donc ajouter quatre branches supplémentaires dans notre hiérarchie (voir figure 9.4).

9.3.2.3 Une hiérarchie pour les méthodes d'optimisation

La hiérarchie que nous avons présentée concernant la méthode de traitement du problème multiobjectif n'est pas suffisamment discriminante pour permettre une bonne classification des méthodes d'optimisation. En effet, il existe beaucoup de méthodes dans la catégorie des méthodes couplées. Il serait commode de pousser un peu plus loin la hiérarchie dans cette catégorie.

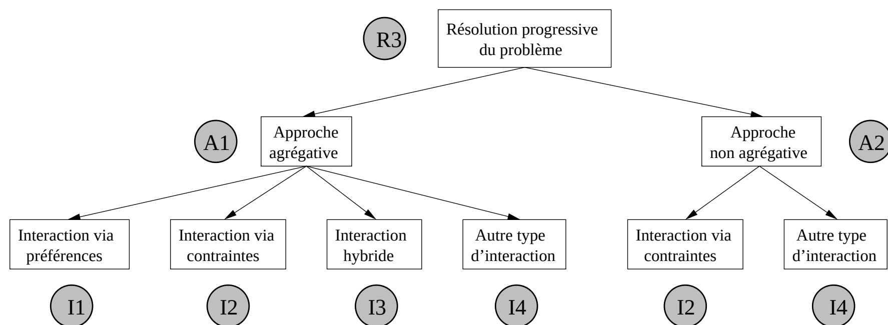


FIG. 9.4 – La hiérarchie des modes d'interaction.

Lorsque l'on regarde les méthodes utilisées dans ce domaine, on constate qu'elles sont de trois types :

- Les méthodes d'amélioration par voisinage :
  ces méthodes partent d'un point initial et l'améliorent par une “analyse” de tout ou partie du voisinage du point courant. Dans cette famille, ou trouve le recuit simulé et la recherche locale, par exemple.
- Les méthodes évolutionnaires :
  ces méthodes vont transformer des individus d'une population de manière à ce qu'ils se rapprochent d'une zone optimale ou d'un point optimal.
- Les autres méthodes :
  ce sont des méthodes qui sont peu utilisées dans le domaine de l'optimisation multiobjectif. On trouve, par exemple :
  - Les méthodes de résolution par construction :
    ces méthodes affectent une valeur définitive à une ou plusieurs variables à une itération donnée. Elles ne reviennent jamais sur une décision. On peut citer comme exemple de ces méthodes la “méthode glouton”.
  - Les méthodes de résolution par décomposition :
    ces méthodes peuvent, par exemple, décomposer l'ensemble des solutions d'un problème en une arborescence et parcourir cette arborescence à la recherche de la meilleure solution. Un exemple de ce type de méthode est la méthode par séparation et évaluation (*Branch and Bound*).

Il n'est pas intéressant de poursuivre cette décomposition plus loin, car cela ne ferait qu'alourdir la hiérarchie et, comme nous l'avons précisé plus haut, au vu des méthodes utilisées dans le domaine de l'optimisation multiobjectif, cette démarche n'est pas fructueuse.

Le dernier niveau hiérarchique que nous allons ajouter concerne les méthodes évolutionnaires. En effet, dans ces méthodes, on peut distinguer deux grandes familles :

- Les méthodes évolutionnaires élitistes :
  ces méthodes ont une archive dans laquelle sont sauvegardés les meilleurs individus rencontrés au cours de l'optimisation. Les individus de cette archive peuvent être transmis intacts à l'itération suivante ou alors intervenir de manière à améliorer la recherche. La méthode PASA est un exemple de méthode de cette famille.
- Les méthodes évolutionnaires non élitistes :
  ces méthodes n'ont pas d'archive et sont généralement plus simples à mettre en œuvre. La méthode MOGA en est un exemple.

La hiérarchie concernant plus spécifiquement les méthodes d'optimisation est représentée à la figure 9.5.

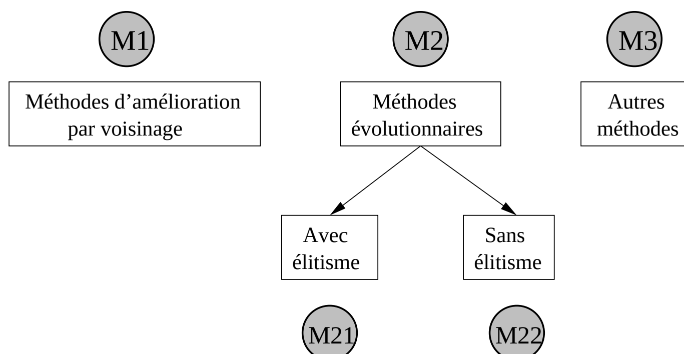


FIG. 9.5 – Une hiérarchisation supplémentaire au niveau des méthodes d’optimisation.

9.3.2.4 Comment exploiter cette hiérarchie

C’est à ce moment qu’un problème survient. En effet, sans une connaissance minimale du problème et des méthodes, il est très difficile de choisir une méthode d’optimisation multiobjectif.

Lorsque l’on choisit une méthode d’optimisation multiobjectif, plusieurs facteurs entrent en ligne de compte :

- Des exigences quant à la qualité de la solution :
  certains problèmes ont un optimum très difficile à trouver. Ces problèmes seront très probablement mieux résolus par une métaheuristique du type recuit simulé. Pour ces problèmes, on peut être prêt à passer plusieurs heures ou plusieurs jours à rechercher une solution. C’est le cas de problèmes stratégiques, tels que celui de l’optimisation de structure, où une structure avec une quantité moindre de matériau, mais répondant toujours au cahier des charges, pourra faire économiser une somme importante sur le coût de l’ouvrage.
  Pour d’autres problèmes, moins stratégiques, on pourra accepter une “solution” qui ne sera qu’une amélioration du point initial. On pourra alors choisir une méthode de type descente, ou un recuit simulé avec une décroissance rapide de la température.
- Une structure particulière de l’espace de recherche :
  la présence d’optima locaux va influer sur le choix de la méthode d’optimisation. Il faut admettre que, sur certains problèmes très difficiles, il est impossible de connaître à l’avance la proportion d’optima locaux et donc de juger de la difficulté a priori du problème.
  L’aspect convexe ou non convexe de la surface de compromis va influer sur le choix de la méthode de traitement multiobjectif. Là aussi, il est très difficile de juger de l’allure de la surface de compromis a priori, et donc de choisir une méthode de traitement multiobjectif adaptée au problème.
  Pour un problème convexe, on choisira une méthode de pondération des fonctions objectif, alors que, pour un problème non convexe, on choisira plutôt une méthode de Tchebychev.
- La capacité à exprimer précisément ses préférences :
  si les préférences sont bien définies, on choisit une méthode de résolution de type *a priori*. En effet, il faut bien avoir à l’esprit que chercher à approcher une surface de compromis est une tâche qui peut être longue et difficile.
  Si les préférences ne sont pas bien définies, on a deux solutions :

- Soit aider le décideur à préciser ses préférences par l'intermédiaire d'une résolution *progressive* du problème. Cette méthode de résolution n'a pas les faveurs des utilisateurs, car elle mobilise le décideur tout au long de la recherche. De plus, la durée qui sépare deux étapes d'interaction peut être longue.
- Soit approcher la surface de compromis en engendrant un nombre fini de solutions bien réparties sur celle-ci : c'est l'approche *a posteriori*. Cette approche de la résolution d'un problème multiobjectif est commode, surtout quand le circuit de décision est long et/ou mal identifié. Elle laisse ainsi l'occasion au(x) décideur(s) de juger les solutions proposées "sur pièce", par comparaisons successives, ou d'extraire une solution en utilisant une méthode d'aide à la décision.

Il est clair que si le décideur a déjà ces réponses en main, la sélection d'une méthode n'en sera que plus aisée. Il faut donc voir cette hiérarchie plus comme un outil d'aide au questionnement qu'un outil d'aide à la sélection.

9.3.3 Classification de quelques méthodes

Nous allons maintenant classer les méthodes traitées dans ce livre en utilisant la hiérarchie que nous venons de déterminer.

Les méthodes scalaires

Méthode	Trajet dans la hiérarchie
Pondération des fonctions objectif	$R1/A12/C2$
Keeney-Raiffa	$R1/A12/C2$
Distance à un objectif	$R1/A12/C2$
Compromis	$R1/A11/C2$
Hybride	$R1/A13/C2$
But à atteindre	$R1/A11/C2$
But programmé	$R1/A11/C2$
Ordonnancement lexicographique	$R1/A11/C2$
Contraintes d'égalité propres	$R2/A11/C2$
Contraintes d'inégalité propres	$R2/A11/C2$
Lin-Tabak	$R2/A11/C2$
Lin-Giesy	$R2/A11/C2$

Les méthodes interactives

Méthode	Trajet dans la hiérarchie
Compromis par substitution	$R3/I2/C2$
STEP	$R3/I2/C2$
Gandel	$R3/I2/C2$
Jahn	$R3/I4/C2$
Geoffrion	$R3/I4/C2$
Simplex	$R3/I1/C2$

Pour les méthodes de Jahn et de Geoffrion, on remarque que l'interaction se fait via le choix de la longueur d'un pas. Ce mode d'interaction ne permet pas à l'utilisateur de modéliser simplement ses préférences.

Les méthodes floues

Méthode	Trajet dans la hiérarchie
Sakawa	$R1/A12/C2$
Reardon	$R1/A12/C2$

Les méthodes d'aide à la décision

Méthode	Trajet dans la hiérarchie
ELECTRE I	$R1/A12/C2$
ELECTRE IS	$R1/A12/C2$
ELECTRE II	$R1/A12/C2$
ELECTRE III	$R1/A12/C2$
ELECTRE IV	$R1/A12/C2$
ELECTRE TRI	$R1/A12/C2$
PROMETHEE 1	$R1/A12/C2$
PROMETHEE 2	$R1/A12/C2$

Les méthodes à base de métaheuristique

Méthode	Trajet dans la hiérarchie
PASA	$R\{1,2\}/A12/C1/M1/E1$
MOSA	$R\{1,2\}/A12/C1/M1/E2$
MOGA	$R2/A2/C1/M2/E2$
NSGA	$R2/A2/C1/M2/E2$
NPGA	$R2/A2/C1/M2/E2$
WARGA	$R2/A2/C1/M2/E2$

Nous avons classé ces méthodes dans les catégories $R\{1,2\}$ car, dans leur utilisation courante, ces méthodes sont plutôt $R2$, mais, en théorie, elles seraient plutôt $R1$. La raison en est qu'elles exploitent l'effet stochastique du recuit simulé.

9.3.4 Comment choisir une méthode

Nous allons maintenant illustrer le choix d'une méthode d'optimisation à travers deux exemples :

– Le dimensionnement d'une poutre (voir section 1.6.3).
  Ce problème est bien défini. Nous avons deux fonctions objectif : minimiser la section de la poutre et minimiser la déformation de la poutre.
  Nous savons a priori que ce problème n'est pas difficile. Il n'a pas d'optima locaux et la structure de la surface de compromis est convexe.
  Cette poutre intervient dans une structure plus complexe (un pont par exemple). Nous voulons donc pouvoir tester plusieurs solutions pour simuler le comportement global de la structure complexe en fonction de la solution choisie pour la forme de la poutre.

– Le dimensionnement d'un réseau de télécommunications (voir chapitre 11).
  Pour ce problème, nous avons plusieurs fonctions objectif : le coût total engendré par la mise en place de l'infrastructure du réseau et la disponibilité du réseau.
  Ce problème est un problème d'optimisation combinatoire a priori difficile.
  Il est possible de formuler un compromis entre les deux fonctions objectif.

Exemple 1 : le dimensionnement de la poutre

Premier niveau dans la hiérarchie : quel mode de résolution choisir ?

Nous voulons obtenir une approximation de la surface de compromis, car nous voulons avoir plusieurs solutions en main, pour les tester sur une structure plus complexe. Il s'agit donc d'un mode de résolution du type *a posteriori*.

Second niveau dans la hiérarchie : quel mode de traitement du problème multiobjectif choisir ?

A ce niveau, nous avons le choix. Nous allons choisir un mode de traitement non agrégatif, car nous avons à l'esprit la propriété suivante de ce mode de traitement : il n'y a pas de préférence à déterminer.

Troisième niveau dans la hiérarchie : quel type de couplage vis-à-vis de la méthode d'optimisation ?

Nous allons utiliser une méthode du type algorithme génétique, car nous avons à notre disposition un algorithme génétique déjà implémenté. Nous choisissons donc une méthode couplée.

Quatrième niveau dans la hiérarchie : quel type de méthode ?

Nous avons choisi une méthode évolutionnaire.

Cinquième niveau dans la hiérarchie : quel type d'élitisme ?

Là encore, nous avons le choix. Notre problème est simple à résoudre. Nous choisissons une méthode sans élitisme car nous ne voulons pas perdre de temps à développer un module d'élitisme pour résoudre un problème aussi simple.

Le trajet dans la hiérarchie est donc le suivant :

$$R2/A2/C1/M2/E2$$

Choix final :

Nous avons donc le choix entre :

MOGA, NSGA, NPGA et WARGA.

Le choix final se portera sur la méthode NSGA, car elle permet d'obtenir une bonne répartition des solutions sur la surface de compromis, sans requérir de réglages trop complexes.

Exemple 2 : le dimensionnement d'un réseau de télécommunications

Premier niveau dans la hiérarchie : quel mode de résolution choisir ?

Pour ce problème, nous avons eu la possibilité de définir des préférences vis-à-vis des fonctions objectif. Nous cherchons donc une solution en particulier. Le mode de résolution retenu est la résolution *a priori*.

Second niveau dans la hiérarchie : quel mode de traitement du problème multiobjectif choisir ?

Nous ne connaissons pas a priori l'allure de la surface de compromis. Il nous faut donc choisir une méthode qui soit efficace sur des problèmes dont la surface de compromis est non convexe (cas le plus défavorable).

Étant parti sur un mode de résolution *a priori*, nous allons nous orienter vers un mode de traitement agrégatif.

Troisième niveau dans la hiérarchie : quel type de couplage vis-à-vis de la méthode d'optimisation ?

A ce niveau, nous avons le choix. Nous allons utiliser, dans un premier temps, une méthode découplée. Si la résolution du problème n'est pas satisfaisante, nous recommencerons, mais plutôt avec une méthode couplée.

Quatrième niveau dans la hiérarchie : quel type de méthode ?

La littérature nous informe que les meilleurs résultats ont été obtenus avec des méthodes du type amélioration par voisinage (du fait que ces méthodes sont plus faciles à implémenter que des méthodes évolutionnaires). De plus, dans un mode de résolution *a priori*, il n'est pas intéressant d'utiliser une méthode évolutionnaire, car nous cherchons une seule solution.

Cinquième niveau dans la hiérarchie : quel type d'élitisme ?

Compte tenu de la difficulté a priori de la recherche, il va être intéressant de conserver la trace des bonnes solutions déjà rencontrées lors de l'optimisation. Nous choisissons une méthode avec élitisme.

Le trajet dans la hiérarchie est donc le suivant :

$$R1/A12/C2/M1/E2$$

Choix final :

Dans la liste des méthodes que nous avons déterminée, il n'y a pas de méthodes dont le trajet dans la hiérarchie corresponde à celui que nous venons de déterminer. Cependant, si nous faisons abstraction de l'élitisme, nous pouvons choisir une méthode de traitement du type distance de Tchebychev couplée à une méthode d'optimisation. Pour satisfaire nos recommandations en terme de choix de méthode, il nous suffira alors d'implémenter un module d'archivage, pour conserver la trace des bonnes solutions déjà rencontrées, ou alors de poursuivre notre recherche de méthode à l'aide d'une analyse bibliographique.

Troisième partie

Etudes de cas

Chapitre 10

Etude de cas n°1 : qualification de code scientifique[1]

10.1 Introduction

Lorsque l'on développe un code scientifique dédié à la simulation d'un processus industriel (chaîne de distillation chimique (voir le lien [Ascend] vers le logiciel Ascend de simulation mathématique), production d'énergie au sein d'un réacteur nucléaire, etc.), on ne peut pas prendre en compte tous les paramètres de ce processus. L'ingénieur doit alors réduire la complexité du modèle du processus pour qu'il puisse être simulé. Les différentes approximations qu'il est amené à faire entraînent souvent un écart plus ou moins important entre les résultats obtenus par simulation et les résultats réels. On dispose alors d'un "artifice" qui permet de combler cet écart : le calage du code de simulation sur les résultats réels. Ce calage se fait par action sur certains paramètres du modèle de manière à ce que les résultats calculés correspondent mieux aux résultats réels.

10.2 Description du problème

Nous décrivons le problème à l'aide d'un exemple. Le domaine physique concerné est celui de la thermo-hydraulique diphasique.

Les données expérimentales sont les mesures de "taux de vide" en dix points répartis uniformément le long de la section d'un canal dont une des parois est chauffée. Ce canal est le siège d'un écoulement d'eau sous forme d'un mélange liquide vapeur. Un exemple de courbe que l'on peut obtenir est représenté à la figure 10.1. Le taux de vide est calculé par un code dont un des modèles mis en œuvre pour simuler l'écoulement dépend de deux paramètres susceptibles de pouvoir modifier l'allure de la courbe obtenue en sortie : C_1 et C_∞. Le but du calage sera alors de trouver une ou plusieurs valeurs du couple de paramètres (C_1, C_∞) telles que la courbe expérimentale soit la plus voisine possible de la courbe calculée.

Le problème qui survient est alors le suivant : que veut dire "le plus voisin de" :

– des écarts nuls en tous les points ?
– un écart nul sur le point dont le niveau est le plus important (le point n°10 sur l'exemple de la figure 10.1) appelé "point chaud" ?
– des écarts uniformément répartis ?

[1] Ce chapitre a été rédigé en collaboration avec M. Dumas du CEA (michel.dumas@cea.fr).

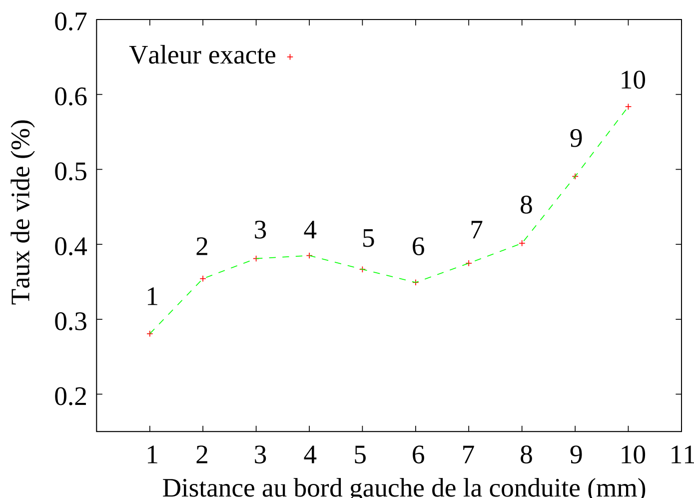


FIG. 10.1 – Une mesure du taux de vide sur les dix points de mesure.

– un écart quadratique moyen minimal ?
– ...

Comme on peut le voir, le choix du type de définition du “voisinage” est important et conditionne de manière importante le calage.

Une autre approche peut être menée où l’on considère les écarts absolus entre calcul et expérience sur chaque point comme des fonctions objectif. Dans ce cas, la faible discrimination que l’on ressent entre les différentes définitions énumérées plus haut apparaît sous un nouveau jour. Ces définitions peuvent être vues comme des solutions particulières d’un problème multiobjectif. Ces solutions seront plus ou moins équivalentes, car elles feront partie de la surface de compromis et seront non dominées.

Proposer l’étape du calage comme une étape d’optimisation multiobjectif va permettre de clarifier le problème : compte tenu du nombre de fonctions objectif mis en œuvre, il n’existe pas de solution unique à ce problème (c’est un problème multiobjectif) mais une infinité et, parmi ces solutions, plusieurs vont se distinguer :

– On trouvera des couples (C_1, C_∞) permettant d’obtenir des valeurs moyennes très précises.
– On trouvera des couples (C_1, C_∞) permettant de simuler avec précision la valeur du point chaud. Ces couples peuvent être très pratiques pour le dimensionnement de la conduite.
– ...

Une différence importante est à souligner par rapport à la démarche classique consistant à définir *a priori* une formule de composition des fonctions objectif puis à effectuer le calage en utilisant cette formule. L’approche multiobjectif produit, dans le meilleur des cas, toutes les solutions, et laisse à l’utilisateur le soin de faire un choix parmi ces solutions. Il pourra favoriser certains comportements du simulateur plutôt que d’autres, ces comportements répondant à un besoin particulier (précision accrue sur certaines valeurs). L’utilisateur doit faire un compromis pour choisir une solution et il est alors plus conscient que ce compromis pourra avoir une influence sur les résultats de la simulation.

10.3 Représenter la surface de compromis

Le problème qui intervient maintenant va être celui de la représentation des solutions. Idéalement, il faudrait représenter toute la surface de compromis. Nous rappelons que la définition de la surface de compromis est la suivante :

> La surface de compromis est l'ensemble des points tels que si l'on améliore un ou plusieurs critères on obtient alors une détérioration des autres critères. (voir section 1.6)

La difficulté, dans ce cas, est que celle-ci doit être représentée, pour notre exemple, dans un espace à dix dimensions. Il faut alors trouver une autre solution.

Cette autre solution peut être de présenter à l'utilisateur la zone de Pareto, qui est l'image de la surface de compromis dans l'espace des paramètres (voir figure 10.2). L'avantage de cette approche est que l'espace des paramètres est de dimension 2, donc "facilement" représentable. Le problème qui se pose maintenant est que, pour obtenir une bonne description de

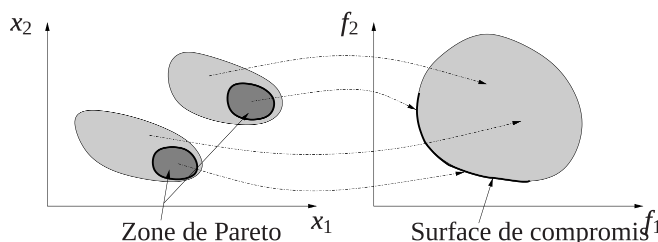


FIG. 10.2 – La correspondance entre la surface de compromis et la zone de Pareto.

la surface de la zone de Pareto, il faut un très grand nombre de calculs de fonctions objectif et que, compte tenu de la durée d'une simulation, cette masse de calculs n'est pas envisageable.

La solution va être de changer de simulateur. Pour pouvoir produire tous ces calculs, nous allons utiliser un approximateur neuronal (voir [Dreyfus et al. 01]) qui présentera l'avantage de produire des résultats de simulation presque équivalents à ceux du simulateur classique, à une marge d'erreur près, bien sûr.

Pour "entraîner" l'approximateur neuronal, on procède de la manière suivante :

Etape 1 : Les paramètres C_1 et C_∞ évoluent dans un domaine prédéfini :

$$0 \leq C_1 \leq 2 \text{ et } 0.5 \leq C_\infty \leq 2$$

On tire 1 500 points distincts dans cet espace des paramètres.

Etape 2 : On effectue une simulation pour chacun de ces points.

Etape 3 : On utilise ces données (deux entrées, C_1 et C_∞, dix sorties, les mesures du taux de vide en chacun des points de la section de la conduite) pour entraîner un approximateur neuronal (voir [Dreyfus et al. 01] pour plus de détails sur la façon d'entraîner un approximateur neuronal).

De cette manière, on détermine dix approximateurs neuronaux : un par point de mesure. Sur les figures 10.3, 10.4 et 10.5, on peut voir des exemples de types de résultats que l'on obtient en suivant cette démarche.

Ces figures représentent les réponses des réseaux de neurones lorsque l'on présente en entrée des valeurs de C_1 et C_∞ couvrant tout l'espace des paramètres. Les trois approximateurs choisis correspondent aux points de mesure 1, 7 et 10. Ils ont un comportement représentatif des dix mesures et les valeurs absolues de leurs écarts avec des valeurs expérimentales définissent les trois fonctions objectif du problème.

Comme on peut le voir en observant ces trois courbes, quand on déplace un point dans l'espace des paramètres, certaines fonctions objectif sont antagonistes. Certaines augmentent pendant que d'autres diminuent. C'est une caractéristique importante d'un problème multiobjectif.

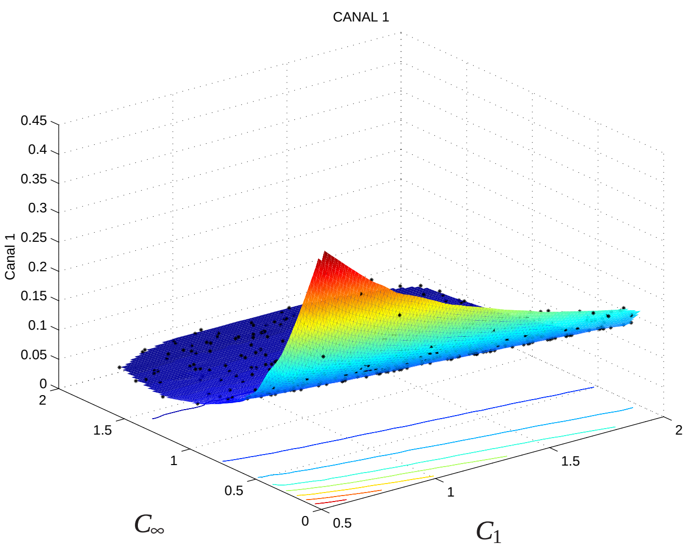


FIG. 10.3 – Réponse de l'approximateur neuronal pour le point de mesure 1.

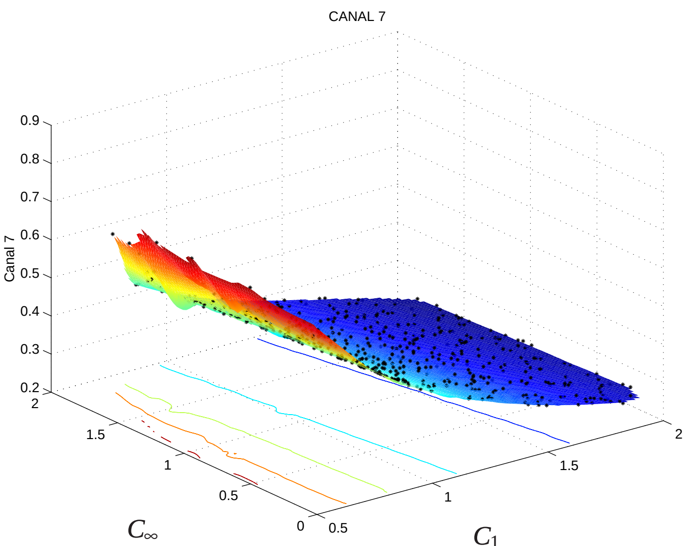


FIG. 10.4 – Réponse de l'approximateur neuronal pour le point de mesure 7.

Il nous faut maintenant rechercher la ou les zones de l'espace des paramètres qui contiennent les solutions Pareto-optimales : la zone de Pareto. Pour cela, on peut tracer ces courbes sur un plan en ne représentant que les courbes de niveaux dont les valeurs correspondent aux mesures expérimentales. A la figure 10.6, on n'a représenté que les courbes de niveau des points

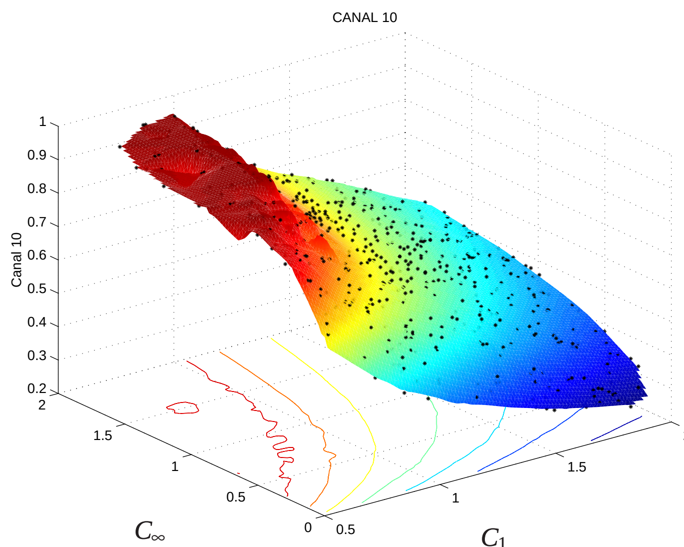


FIG. 10.5 – Réponse de l'approximateur neuronal pour le point de mesure 10.

de mesure 1, 7 et 10. Chaque courbe définit l'ensemble des valeurs des couples (C_1, C_∞) permettant d'obtenir, pour le point de mesure correspondant, un écart nul avec le calcul. Ces trois courbes définissent dans l'espace des paramètres le domaine dont l'image est la surface de compromis. Le domaine A définit l'image de la surface de compromis dans l'espace des paramètres. En effet, dans cet espace, si l'on déplace un point x vers la courbe de niveau minimal du point 10, alors on augmente les valeurs des courbes de niveau des points 7 et/ou 1. De même, si l'on déplace ce point vers la courbe de niveau minimal du point 1, alors on augmente les valeurs des courbes de niveau 7 et/ou 10.

Ce comportement est typique d'un point appartenant à la surface de compromis. A la figure 10.7, on a représenté la zone de Pareto. Dans cette zone, nous avons choisi trois points (A, B et C) correspondant à des solutions Pareto-optimales (car choisies dans la zone de Pareto). A la figure 10.8, nous avons représenté les profils de courbes correspondant à ces trois points. Ces profils donnent tous le même écart par rapport à la courbe expérimentale. En revanche, certains profils vont donner des résultats précis sur les bords de la conduite et pas au centre, d'autres vont donner des profils dont la forme ressemblera au profil expérimental, mais avec un écart maximal plutôt faible.

10.4 Conclusion

Cette démarche peut être généralisée par l'utilisation d'un algorithme génétique multiobjectif. En effet, ce type de méthode se prête bien à l'approximation de la zone de Pareto via l'utilisation d'approximateurs neuronaux (pour la rapidité de simulation).

On peut utiliser un algorithme NSGA avec une gestion de la diversité des solutions dans l'espace des paramètres ou un algorithme génétique à codage réel (voir [Perrin et al. 97]). De cette manière, avec une population relativement petite, on peut espérer obtenir une approximation de la zone de Pareto rapidement.

La principale liberté que permet cette méthode est le recalibrage dynamique en fonction des besoins de simulation de l'utilisateur.

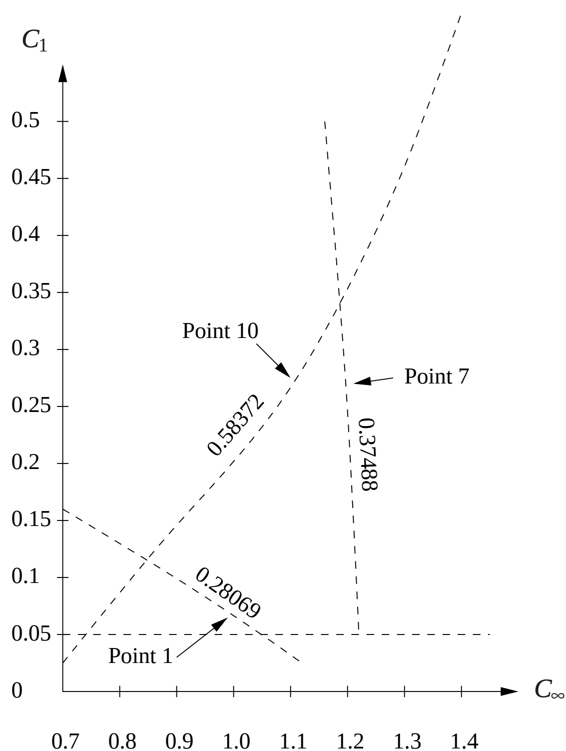
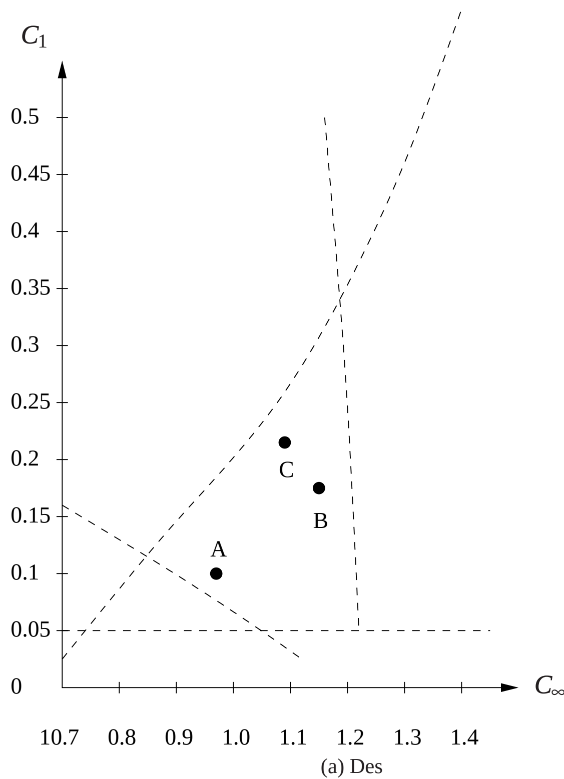
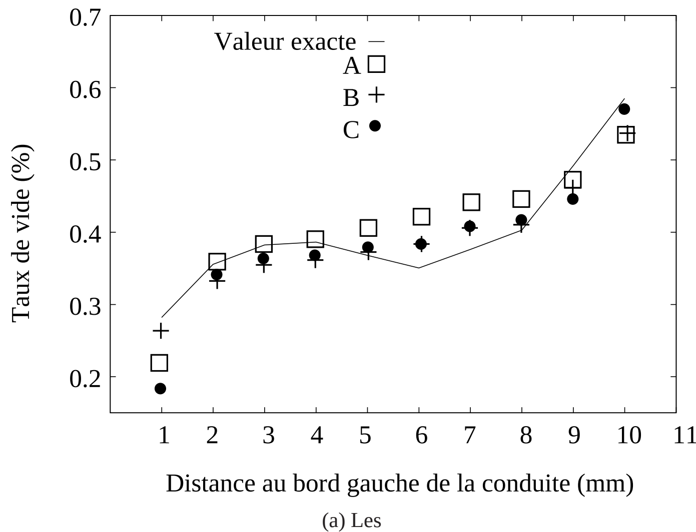


FIG. 10.6 – Les courbes de niveaux minima des points de mesure 1, 7 et 10.



(a) Des

FIG. 10.7 – Des points de la zone de Pareto.



(a) Les

FIG. 10.8 – Les courbes correspondant aux points de la zone de Pareto.

Chapitre 11

Etude de cas n°2 : étude de l'extension d'un réseau de télécommunications [1]

Un réseau de services est en perpétuelle mutation. A partir du moment où il est opérationnel, il n'est qu'au début de sa vie. Le réseau va évoluer au fil de l'accroissement de la demande qu'il doit acheminer d'un point à un autre. Son évolution se fera aussi en fonction de l'obsolescence des équipements qui le constituent, mais surtout au rythme des décisions stratégiques de l'opérateur qui l'exploite.

Ces décisions stratégiques impliquent la plupart du temps des investissements. A ce titre, elles doivent être judicieusement prises. Toutefois, leur dépendance à plusieurs éléments, parfois contradictoires, parfois très corrélés, rend ces choix stratégiques d'évolution extrêmement complexes.

Soit un réseau existant, schématisé à la figure 11.1 en gras, et considérons qu'une extension de ce réseau soit envisagée au sud-est de celui-ci (extension modélisée en traits fins).

Avant d'en arriver à une telle décision, de nombreuses études ont dû être menées afin de caractériser la zone géographique que l'on souhaite développer (limitation de la zone, clients potentiels sur la zone, infrastructures existantes sur la zone, situation économique de la zone, etc.), de déterminer les points stratégiques (par leur concentration de clients et/ou les infrastructures déjà existantes, etc.), de déterminer les liens les plus intéressants ou faciles à ouvrir entre ces nœuds, de caractériser les fournisseurs présents sur le marché (type d'équipements qu'ils proposent, leur adéquation avec le réseau existant, délais de livraisons, coûts, etc.), etc. Ce problème est donc extrêmement complexe, et de nombreuses études économiques débutent sa résolution.

Le problème multicritère présenté ici concerne une partie de ce problème global. Cette étude vise à dimensionner l'extension du réseau proposée : quels sont les investissements les plus rentables, en termes de coût et de disponibilité du réseau, permettant malgré tout au réseau d'acheminer toutes les demandes d'une matrice de demandes cible ?

[1] Ce chapitre a été rédigé par C. Decouchon-Brochet, France Télécom - Recherche et Développement, (camille.decouchonbrochet@francetelecom.com)

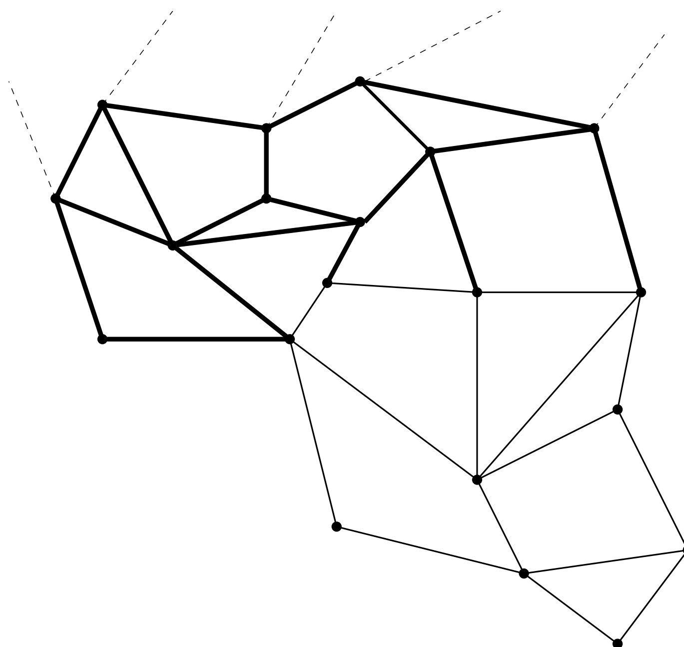


FIG. 11.1 – Réseau existant et extension envisagée.

11.1 Réseau

Un réseau est modélisé par un graphe G non orienté, dont les éléments x sont les nœuds ($x \in N$) et les arcs ($x \in A$). Chacun de ces éléments est caractérisé par sa capacité c_x et son coût d’installation $C(x, c_x)$.

La fonctionnalité du réseau étant de faire transiter de l’information, les caractéristiques de ces informations doivent être spécifiées. Ces caractéristiques sont spécifiées sous forme d’un ensemble de demandes (matrice de demandes) correspondant chacune à une certaine ***quantité*** devant être acheminée d’un nœud ***origine*** vers une ***destination*** (le routage).

Nous considérons que ces demandes ne sont pas concaténées, c’est-à-dire qu’elles peuvent être sécables et donc transmises de l’origine à la destination par plusieurs chemins.

D’autre part, les arcs sont bidirectionnels, c’est-à-dire qu’ils peuvent faire transiter de l’information indifféremment dans un sens ou un autre.

11.2 Critères de choix

11.2.1 Paramètres permettant la modélisation de la disponibilité

La disponibilité d’un réseau est caractérisée par son aptitude à faire face aux fluctuations des demandes qui doivent y transiter.

Autrement dit, si une partie de la demande est non routée, la disponibilité du réseau s’en voit dégradée. Par exemple :

– La non-sécurisation de tout ou partie du réseau diminue le taux de disponibilité, puisque les cas de panne ne sont pas envisagés.
– Le non-surdimensionnement du réseau réduit sa disponibilité, puisqu’il ne pourra pas faire face à un éventuel accroissement de la demande.

Dans l'étude que nous présentons ici, la sécurisation du réseau, ou des demandes plus précisément, va être étudiée en garantissant en permanence, pour chacune d'elles, deux chemins disjoints, en termes d'arcs et de nœuds, de capacité suffisante sur le réseau. Ceci permettra ainsi de faire face aux occurrences de pannes.

Le paramètre qui va modéliser la disponibilité est le taux de demandes sécurisées : T.

Ainsi plus la valeur de ce paramètre se rapprochera de 1, plus la disponibilité du réseau sera considérée comme satisfaisante.

11.2.2 Lien entre la disponibilité et le coût

Bien entendu, la considération de tous ces points garantissant une bonne disponibilité du réseau est limitée par le coût que leur mise en place implique. Si deux chemins disjoints sont garantis pour chaque demande, cela implique un surdimensionnement du réseau qui sera forcément coûteux. A l'inverse, réduire la disponibilité offerte par le réseau permettrait d'avoir un coût en infrastructures du réseau qui soit plus faible.

De manière générale, le problème de la disponibilité consiste donc à trouver un compromis entre le coût global d'investissement et la disponibilité du réseau résultant, et à chercher s'il est intéressant de payer le prix fort en investissement pour garantir une disponibilité extrêmement élevée, ou s'il ne vaut pas mieux payer des pénalités en cas d'incident qui saturerait le réseau, en des points où la disponibilité est normalement garantie.

Dans le paragraphe qui suit, nous décrivons les coûts intervenant dans cette étude.

11.2.3 Coûts

Les coûts apparaissant dans l'étude du déploiement d'un réseau peuvent être classés en trois parties.

11.2.3.1 Les coûts d'investissement

Ce sont les coûts liés aux travaux d'infrastructure du réseau, sachant que ces travaux peuvent être de types différents :
- ajout d'un arc ;
- ajout d'un nœud ;
- modification d'un arc (extension de la capacité de l'arc, etc.) ;
- modification d'un nœud (changement d'équipement en ce nœud pour des équipements de nouvelle génération, etc.) ;
- etc.

Pour toutes ces modifications possibles, il existe plusieurs sortes de coûts :
- le coût lié aux études préalables ;
- le coût d'achat des équipements ;
- le coût des travaux d'installation (coût lié à l'activité humaine).

11.2.3.2 Les coûts de maintenance/gestion

Ces coûts sont ceux qui permettent de faire vivre le réseau. Il s'agit :
- du coût de réaménagement de réseau (permettant par exemple de mettre à jour les routages obsolètes du fait de l'évolution du réseau) ;
- du coût de maintenance préventive ;
- du coût de maintenance curative :
  - diagnostic ;

- réparation.

Là aussi, ces coûts sont composés de coûts d'achat d'équipements et de coûts d'activité humaine.

11.2.3.3 Les coûts liés à l'activité économique du réseau (rentabilité, pénalités)

Ce coût regroupe les différents coûts liés à l'activité économique du réseau.

L'extension du réseau nécessite des coûts d'investissement, mais elle permettra, en outre, au réseau de router de nouvelles demandes (ayant d'autres origines et/ou destinations) et donc de servir de nouveaux clients. Des **bénéfices** peuvent donc apparaître à ce niveau là.

Ces bénéfices peuvent cependant être diminués par des coûts de **pénalités** provenant d'une mauvaise qualité de service à un moment donné (disponibilité garantie non tenue, etc.).

11.3 Etude d'une extension du réseau

11.3.1 Problématique

Comme nous l'avons vu au début de cet exemple, l'opérateur souhaite étendre le réseau qu'il exploite sur une zone voisine.

Le sous-réseau correspondant à cette extension est représenté à la figure 11.2.

Les nœuds de ce sous-réseau à créer sont connus.

Des liens sont proposés.

Il s'agit de déterminer si ces liens doivent être mis en service, et dans l'affirmative, avec quelle capacité. Bien entendu, la mise en service de chaque lien a un coût qui croît avec sa capacité.

D'où la définition d'une matrice contenant le coût de mise en service d'un arc a de capacité c_a : $C(c_a)$. Les coûts liés à l'activité du réseau ayant été évalués et additionnés au coût d'investissement, le tout ramené à chaque arc.

11.3.2 Modélisation

11.3.2.1 Variables

Les variables à considérer sont donc :

- la valeur de la capacité de chaque arc a : c_a ;
- le coût de mise en service d'un arc a, de capacité c_a : $C(c_a)$;
- le paramètre qui caractérise la disponibilité du réseau : T.

On notera C_T le coût total d'investissement, égal à la somme des coûts des arcs installés.

$$C_T = \bigoplus_{i=1}^{N_a} C(c_i) \tag{11.1}$$

- i désignant les arcs,
- et N_a étant le nombre d'arcs du réseau d'extension.

11.3.2.2 Critères

Le problème consiste donc à déterminer la combinaison optimale des capacités (variables c_i, $i \in [1, N_a]$) à placer sur les arcs en optimisant au mieux deux paramètres de ce sous-réseau :

- le coût total engendré par la mise en place de son infrastructure ;
- et un paramètre caractérisant la disponibilité du réseau ainsi défini.

$$\begin{cases} \min_{c_i} C_T \\ \max_{c_i} T \end{cases}$$

11.3.2.3 Contraintes

Cette détermination des liens à mettre en service doit être faite en respectant des contraintes :
- le coût total d'investissement sur le réseau doit être inférieur à un seuil maximal :
$$C_T < C_{\text{seuil}}$$
- pour les routages, les capacités des liens sont limitées ;
- le monoroutage de toutes les demandes doit être au minimum garanti ;
- la valeur minimale du paramètre caractérisant la disponibilité du réseau est imposée : 0.5.

11.3.3 Données nécessaires

11.3.3.1 Infrastructure figée du réseau

L'infrastructure dure de l'extension du réseau doit être connue. Dans notre cas :
- le nombre de nœuds est 10 ;
- le nombre d'arcs potentiels est 15 ;
- les caractéristiques (position et lien) de chacun de ces éléments sont décrites plus loin. Sur le graphe de la figure 11.2 les nœuds sont identifiés avec des lettres et les arcs sont numérotés (de 0 à 14).

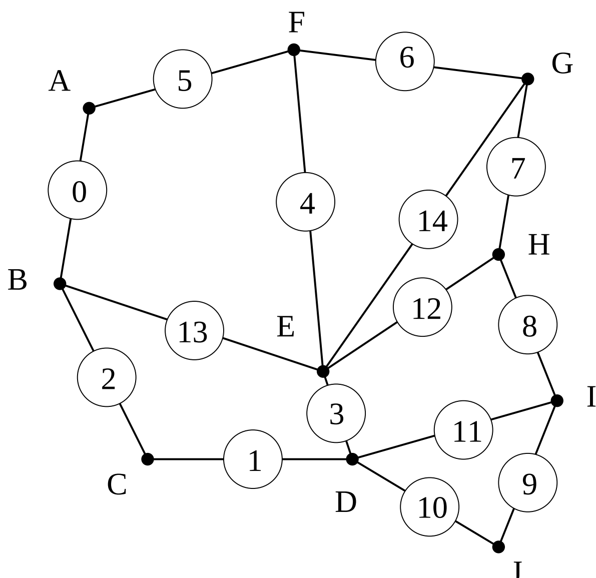


FIG. 11.2 – Sous-réseau à créer.

11.3.3.2 Infrastructure variable du réseau

Les arcs pourront avoir quatre capacités :
- 0 si l'arc est inutile ;

– 16 en unité 155 Mbit/s ;
– 32 en unité 155 Mbit/s ;
– 48 en unité 155 Mbit/s.

11.3.3.3 Matrice de demandes

La matrice de demandes à router sur ce nouveau sous-réseau doit également être connue. Nous considérons quinze demandes représentées dans le tableau 11.1.

Code de la demande	Origine	Destination	Quantité (en unité 155 Mbit/s)
0	G	J	4
1	G	C	8
2	A	I	12
3	B	J	16
4	B	H	4
5	B	I	4
6	F	D	4
7	B	G	8
8	G	A	4
9	A	C	16
10	J	A	12
11	E	H	4
12	E	I	8
13	E	F	4
14	E	C	12

TAB. 11.1 – Matrice de demandes à router.

11.4 Méthodologie de résolution

Le type de problème présenté ici est un problème d'optimisation multicritère, puisque plusieurs critères doivent être optimisés simultanément. La difficulté principale d'un tel problème provient des contradictions entre les différents critères. En effet, dans des cas concrets, il est très rare de trouver une solution qui optimise tous les critères simultanément. Il faut donc trouver un compromis entre ces différents critères, sachant qu'il n'existe pas un seul compromis pour un problème donné, mais plusieurs compromis et que ce nombre de compromis croît avec le nombre de critères. Cette complexité est notamment due :
– à l'importante combinatoire des problèmes de dimensionnement de réseaux ;
– au fait qu'il existe beaucoup de solutions quasi optimales au sens d'un seul critère, ce qui implique un nombre important de compromis.

L'objectif donc, lorsqu'on résout un tel problème, est de trouver un ensemble de solutions non dominées, qui sera ensuite proposé au décideur final, afin que celui-ci établisse son choix.

Notre problème peut être résolu par un algorithme évolutionnaire dont l'ossature est décrite à la figure 11.3. Cet algorithme permettra d'extraire les solutions correspondant aux compromis les plus intéressants. Ce sera ensuite à l'utilisateur de choisir, parmi ces solutions mises en avant, celle qui lui paraît la meilleure.

11.4.1 Définition d'une solution admissible

Une solution admissible sera une combinaison des capacités à placer sur chaque arc $\{c_i \,|\, i \in [1, N_a]\}$ qui permettra à l'infrastructure ainsi définie de respecter les contraintes.

11.4.2 Algorithme

Rappelons que l'organigramme général d'un algorithme génétique est celui représenté à la figure 11.3.

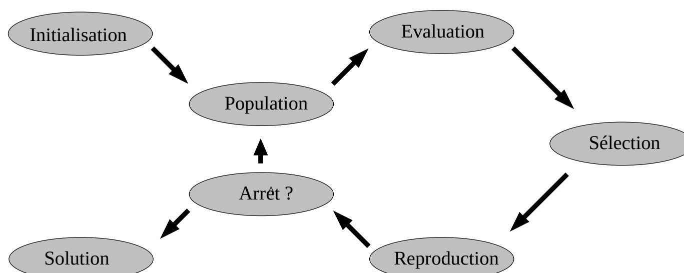


FIG. 11.3 – Organigramme général d'un algorithme génétique.

11.4.2.1 Initialisation 1 : codage

La phase d'initialisation comprend une technique de codage des solutions. Le codage défini ici est très simple puisqu'il est basé sur l'indexation des ensembles de modifications potentielles du réseau.

Une variable par élément nouveau est créée, et la valeur de cette variable précisera quelle capacité est à placer sur l'élément correspondant.

Nous avons vu que quinze arcs pourront être ajoutés au réseau existant pour former le sous-réseau que l'on cherche à créer. Le codage sera donc composé de quinze variables (dans l'ordre de numérotation des arcs). Chacune de ces variables sera :

- nulle si l'arc associé n'est pas créé ;
- égale à 1 si l'arc associé a une capacité (en unité 155 Mbit/s) de 16 ;
- égale à 2 si l'arc associé a une capacité (en unité 155 Mbit/s) de 32 ;
- égale à 3 si l'arc associé a une capacité (en unité 155 Mbit/s) de 48.

Il existe donc $4^{15} = 1\,073\,741\,824$ combinaisons possibles de liens à ouvrir, pour un si petit nombre d'arcs à placer !

11.4.2.2 Initialisation 2 : génération d'une solution initiale

La première phase de l'initialisation est suivie de la création d'une population initiale.

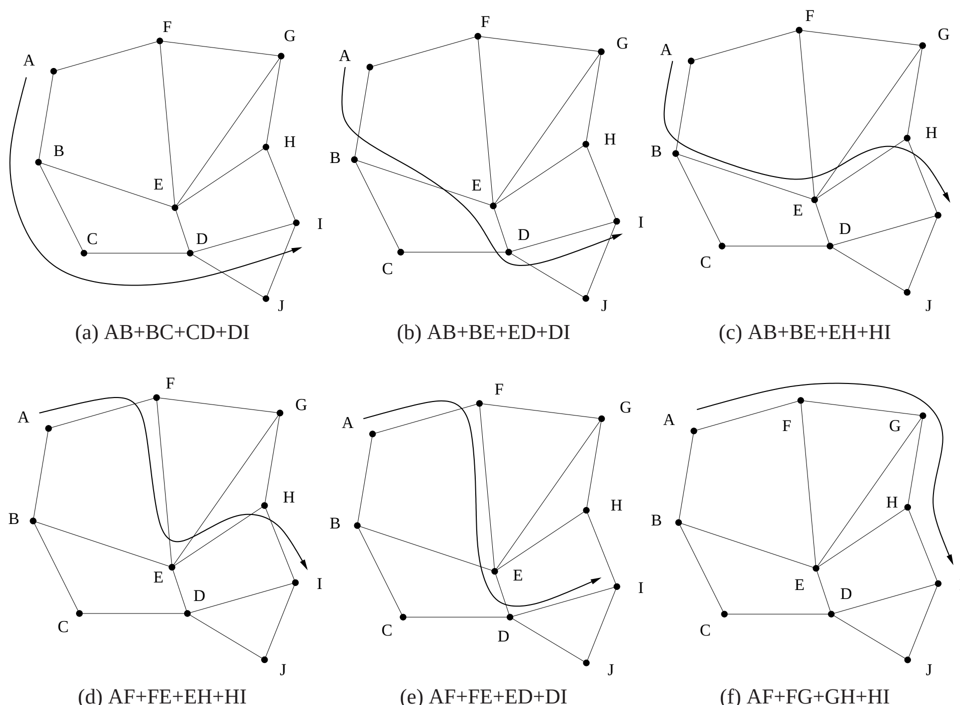
La seule contrainte concernant cette population initiale est qu'elle appartienne au domaine des solutions admissibles.

11.4.2.3 Evaluation 1 : calcul de la solution courante

L'évaluation de la solution va consister d'une part à vérifier que la solution courante est admissible, et d'autre part à trouver la valeur des différents critères que l'on cherche à optimiser.

Pour vérifier que la solution courante est admissible, on va chercher à router, sur le réseau constitué des arcs et capacités de la solution courante, les demandes fournies en entrée. Cela implique donc, à chaque phase d'évaluation de l'algorithme, un routage de plus court chemin de toute la demande.

Ce calcul de routage ne sera pas négligeable dans l'efficacité de l'algorithme puisque, par exemple, pour la demande numéro deux entre A et I, et si tous les arcs sont installés, il existe six routages ayant un nombre de liens minimal (quatre liens). Ces routages sont représentés à la figure 11.4.



(a) AB+BC+CD+DI (b) AB+BE+ED+DI (c) AB+BE+EH+HI

(d) AF+FE+EH+HI (e) AF+FE+ED+DI (f) AF+FG+GH+HI

FIG. 11.4 – Routages de plus court chemin (en nombre de liens) pour une demande entre A et I.

Remarque : ce routage peut être un routage bicritère qui cherche à privilégier les chemins de plus bref délai et de meilleure qualité.

Concernant l'évaluation des critères, il faudra calculer :
- le coût total d'investissement lié à la solution courante,
- la valeur du paramètre représentant la disponibilité du réseau.

Cette seconde évaluation implique que, lors des déterminations de routage de plus court chemin précitées concernant toute la demande, il faudra chercher plus précisément, pour chaque demande, l'existence d'un biroutage sur chemins disjoints (en termes de nœuds et d'arcs), afin de vérifier la disponibilité du réseau correspondant à la solution courante.

Si l'on considère à nouveau la demande numéro deux entre A et I, les deux chemins disjoints pourront être AB+BC+CD+DI et AF+FG+GH+HI. Ceux-ci sont représentés à la figure 11.5.

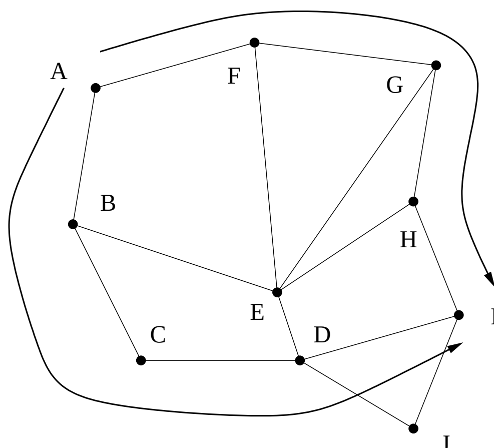


FIG. 11.5 – Exemple de biroutage entre A et I.

Remarque : Si seuls ces arcs (les huit arcs du biroutage précité) constituaient la solution courante, alors cela correspondrait à l'ajout de huit arcs de capacité 16, les autres étant inutiles.

Le codage serait donc : 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0.

Afin d'éviter une énumération exhaustive de tous les biroutages possibles de chaque demande, une heuristique de biroutage de chemins disjoints a été créée. Cette heuristique doit être suffisamment efficace, puisque l'algorithme global s'appuie sur les résultats qu'elle fournit.

D'autre part, pour ne pas chercher systématiquement les biroutages, ceux qui ont déjà été calculés pour un individu donné sont sauvegardés au fur et à mesure dans un dictionnaire.

11.4.2.4 Evaluation 2 : situation de cette solution par rapport au front de Pareto courant

Dans une utilisation multicritère, un individu est d'autant plus performant que le nombre d'alternatives qui le dominent est faible. En pratique, on évalue ce nombre sur un ensemble d'alternatives spécifiques, puisqu'il n'est pas possible de le connaître exactement. Il est fréquent d'utiliser pour cela la population courante de l'algorithme. Dans notre cas, nous avons repris la fonction d'évaluation proposée dans [Fleming 93] qui est la plus économique en temps de calcul (tri complet en $O\left(n^2\right)$). L'évaluation d'un individu est donc effectuée de la manière suivante :

- comparaison de tous les individus entre eux, en sauvegardant le nombre d'individus qui les dominent ;
- renvoi de ces valeurs, la valeur maximale étant attribuée aux individus non dominés ;
- mise à l'échelle pour obtenir la “fitness” finale des individus.

Par le fait que cette évaluation utilise la population courante, les valeurs de performance des individus varient au cours de l'algorithme, ce qui est un inconvénient général, mais incontournable, de ce genre de méthodes.

11.4.2.5 Sélection

Cette phase détermine la manière dont sont sélectionnés les individus qui engendreront la génération suivante. En général, il est préférable de favoriser les individus les plus performants aux dépens des autres. Dans cette optique, plusieurs études récentes [Deb et al. 00c] ont montré l'importance de définir une sélection élitiste. Nous nous sommes appuyés sur ce principe en créant une population archive regroupant les solutions non dominées. Cette population est mise à jour au cours de chaque génération et c'est en son sein que nous effectuons une sélection aléatoire.

11.4.2.6 Reproduction

A partir de la sélection effectuée, la reproduction peut se dérouler. Nous avons utilisé les deux types classiques d'opérateurs élémentaires :
- un opérateur agissant sur un seul individu : *mutation* ;
- un opérateur binaire : croisement ou *crossover*.

L'algorithme doit effectuer un échantillonnage suffisamment large et régulier du front de Pareto. Pour ce faire, les opérateurs définis précédemment ont été associés non seulement à une opération d'***optimisation*** visant à améliorer la valeur de l'individu pour un critère donné, mais également à une opération de ***diversification*** (voir figure 11.6) :
- Opération d'optimisation : cette première opération est semblable à une méthode aléatoire d'optimisation locale. Seules des mutations remplissent ce rôle dans notre algorithme, et une mutation par critère a été définie :
  - Une première mutation améliore la valeur du premier critère (coût total engendré par la mise en place de son infrastructure) en diminuant les indices de modification des gènes, puisque plus ces indices sont faibles, plus la capacité des arcs associés est faible, et donc plus le coût d'infrastructure sera faible.
  - La seconde mutation tend à améliorer la disponibilité du réseau en ajoutant des chemins de routage et en augmentant la capacité des arcs saturés.

  Comme les critères qu'elles tentent d'optimiser, ces mutations ont des effets contradictoires.
- Opération de diversification : l'objectif de cette opération est d'élargir et d'échantillonner le plus régulièrement possible le front de Pareto. Pour ce faire, les mutations définies ci-dessus sont nécessaires. Mais, afin que l'algorithme ne converge pas trop tôt, un opérateur de perturbation aléatoire et de croisement a été défini.

Le mécanisme d'utilisation de ces opérateurs et opérations est le suivant :
- sélection des parents ;
- choix aléatoire d'utilisation ou non de l'opérateur de croisement ;
- choix aléatoire d'utilisation ou non d'une mutation : si oui, sélection aléatoire de la mutation ;
- si aucun opérateur n'a été déclenché, un des deux individus est choisi aléatoirement.

Remarque : les choix aléatoires d'utilisation ou non de l'opérateur de croisement et d'utilisation ou non d'une mutation sont respectivement liés à une probabilité de croisement et une probabilité de mutation prédéfinies.

11.4.2.7 Critère d'arrêt

Notre critère d'arrêt est composé de deux critères :
- un premier critère concernant la taille du domaine exploré ;
- un second critère basé sur la "qualité" de la solution courante.

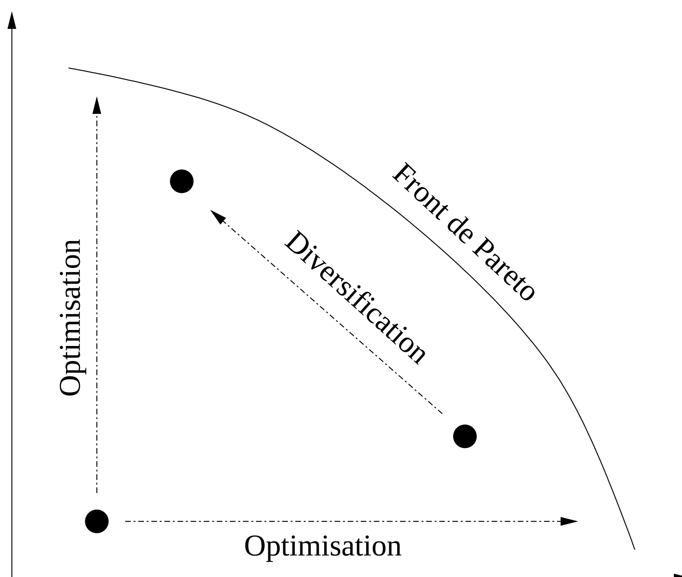


FIG. 11.6 – Différents opérateurs de reproduction.

11.4.2.8 Algorithme global

L'algorithme global utilisé est donc celui de la figure 11.7. Dans cette figure, le terme "dominateur" désigne une solution non dominée.

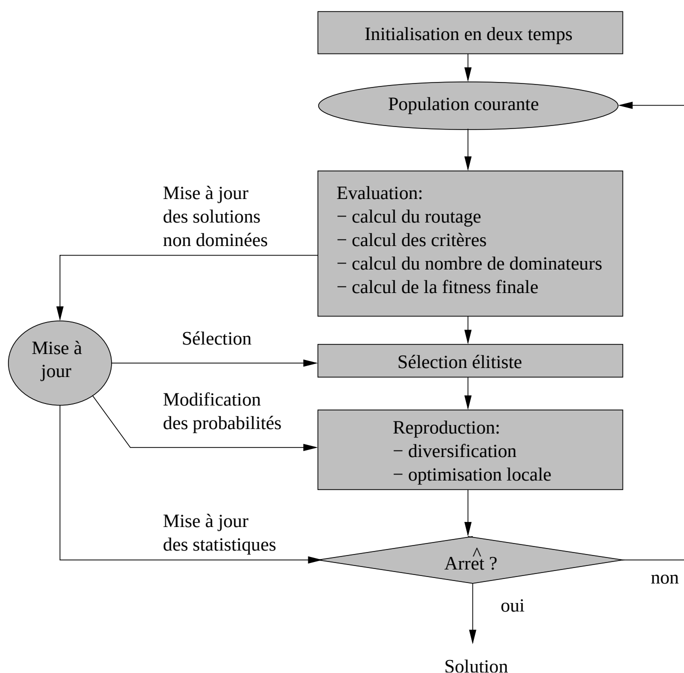


FIG. 11.7 – Organigramme de l'algorithme génétique utilisé.

11.5 Résultats

Quels sont donc les arcs et les capacités de ces arcs qui permettront au réseau d'offrir une bonne disponibilité au moindre coût ?

Les solutions de Pareto obtenues pour ce problème sont au nombre de vingt et un.

Elles sont reportées dans le tableau 11.2.

Codage	Coût	Disponibilité
2 3 2 2 1 1 1 1 1 1 2 1 1 1 2	48944	0
2 3 2 2 1 2 1 1 1 1 2 1 1 1 2	51072	0,1
2 3 2 2 1 2 1 2 2 1 1 1 1 1 2	53056	0,133
2 3 2 2 1 2 1 1 2 1 2 1 1 1 2	53184	0,17
2 3 2 2 1 2 1 1 1 1 2 1 1 1 3	53776	0,23
2 2 2 2 2 2 0 1 3 1 2 1 1 2 2	54336	0,29
2 3 2 2 1 2 1 2 2 1 2 1 1 1 2	55104	0,31
2 3 2 2 1 2 2 1 2 1 2 1 1 1 2	55264	0,37
2 3 2 2 1 2 2 2 2 1 2 1 1 1 2	57184	0,57
2 3 2 3 1 2 2 2 2 1 2 1 1 1 2	59872	0,74
2 3 2 2 1 2 2 2 2 1 2 1 1 1 3	59888	0,77
2 3 2 2 2 2 2 2 2 1 2 1 1 1 3	61680	0,81
2 3 2 2 3 2 2 2 2 1 2 1 1 1 3	63472	0,83
2 3 2 2 2 3 2 2 2 1 2 1 1 1 3	63808	0,84
2 3 2 3 2 2 2 2 2 1 2 1 1 1 3	64368	0,86
2 3 2 3 2 2 2 2 2 2 2 1 1 1 3	65488	0,88
2 3 2 2 2 3 2 2 3 2 2 1 1 1 3	67040	0,89
2 3 2 3 2 3 2 2 2 2 2 1 1 1 3	67616	0,9
2 3 2 2 3 3 2 2 3 1 2 1 1 1 3	67712	0,92
2 3 2 2 3 3 2 2 3 2 2 1 1 1 3	68832	0,93
2 3 2 2 2 3 2 2 3 2 2 1 3 1 3	71872	0,96

TAB. 11.2 – Solutions efficaces obtenues.

On constate logiquement que plus le taux représentant la disponibilité augmente et plus le coût augmente aussi.

Les huit premières solutions ne sont pas admissibles puisque le taux caractérisant la disponibilité du réseau y est inférieur à 0.5. Les solutions Pareto-optimales à traiter par l'utilisateur final sont donc au nombre de treize.

On peut constater également que certains arcs sont très exploités, comme les arcs 1, 3, 5, 8 et 14. D'autres arcs, comme les numéros 9, 11, 12 et 13, sont moins utilisés.

Ces résultats sont reportés sur le schéma de la figure 11.8.

Le front de Pareto apparaît de manière flagrante sur ce graphique.

La partie grisée du graphique recouvre les solutions non admissibles, puisque le coefficient caractérisant la disponibilité du réseau y est inférieur à 0.5.

11.6 Conclusion

L'algorithme utilisé fournit donc de manière assez rapide un front de Pareto suffisamment diversifié et continu. Cet algorithme peut également traiter un troisième critère, correspondant

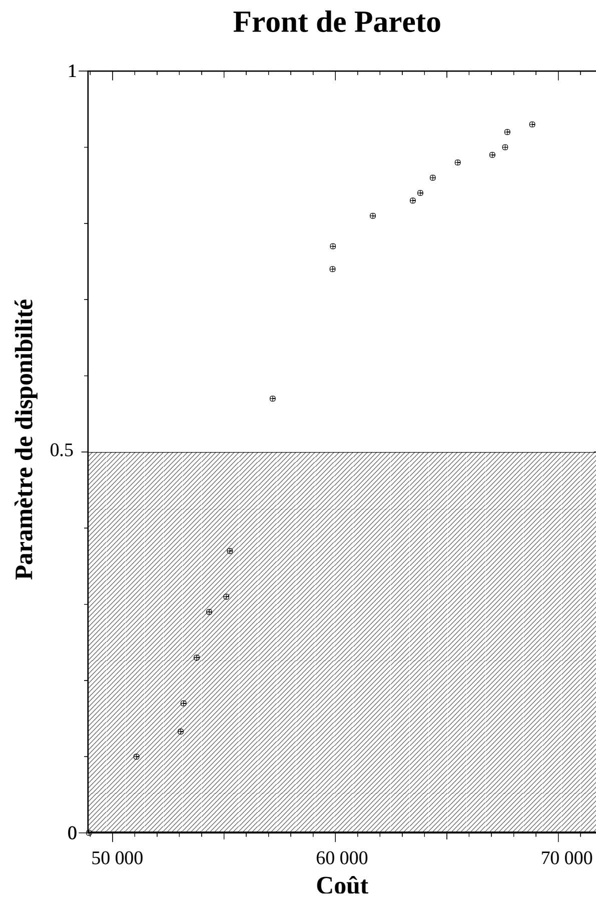


FIG. 11.8 – Ensemble de solutions efficaces.

à la rentabilité de cette extension du réseau. Ce troisième critère aurait été un coût mais qui n'aurait pas forcément évolué de la même manière que notre premier critère, qui est le coût d'investissement.

Bien entendu, la problématique opérationnelle est plus riche que la complexité expliquée par le modèle présenté ici. En effet, une multitude de critères intervient dans la réflexion du décideur : le délai de mise en service du réseau, etc. D'autres critères peuvent, à l'inverse de ceux présentés ici, être non quantifiables mais qualitatifs. Il peut s'agir par exemple de critères stratégiques, en rapport avec la concurrence, ou de critères politiques.

Toutefois, une approche frontale et globale n'est pas réaliste. Le rôle de l'expertise réseau consiste en fait à définir des sous-problèmes algorithmiquement envisageables et suffisamment significatifs, qui apporteront une aide aux décideurs. L'approche présentée ici est donc une facette du soutien possible aux décideurs.

Chapitre 12

Etude de cas n°3 : outils décisionnels multicritères pour le traitement des appels d'offres[1]

12.1 Introduction

EADS LAUNCH VEHICLES (EADS LV), filiale à 100 % du groupe EADS (*European Aeronautic Defence and Space Company*) issu de la fusion le 10 juillet 2000 de Daimler Chrysler Aerospace AG (DASA), de Construcciones Aeronauticas (CASA) et de Aérospatiale Matra, utilise depuis le 1er janvier 1996 de façon opérationnelle un modèle décisionnel multicritère fondé sur la méthode Electre Tri pour répondre ou non aux appels d'offres. Ce modèle, qui sert également pour décider de faire ou non des offres commerciales spontanées, amène à consigner les résultats commerciaux dans une base de données. EADS LV fait plusieurs centaines d'offres commerciales chaque année. Celles-ci sont faites soit de façon non sollicitée, soit dans le cadre d'appels d'offres ou de consultations officielles.

Depuis la création d'un modèle de première génération à neuf critères en 1996, de fréquentes modifications ont été réalisées. Elles ont amené en 2001 à une refonte intégrale du modèle, qui n'était possible qu'avec l'expérience d'une première période et en particulier l'acquisition de données. Afin de prendre en compte la variété des offres commerciales de produits ou de services, le modèle de deuxième génération à cinq critères a conduit à une famille de modèles affinés sur mesure.

Les modèles de première et deuxième génération sont présentés succinctement dans les paragraphes suivants. Pour des raisons évidentes de confidentialité, les valeurs numériques, ainsi que les graphiques, ont été volontairement voilés ou modifiés.

12.2 Modèle de première génération

12.2.1 Mission du modèle

Le premier modèle d'aide à la décision était construit sur la problématique suivante : "Faut-il ou non répondre à cet appel d'offres ?" Il a été rapidement généralisé à la question : "Faut-il faire ou non une offre pour cette affaire nouvelle ?"

[1] Ce chapitre a été rédigé par J.-M. Contant, J.-L. Bussière, G. Piffaretti, EADS Launch Vehicles.

Accepter une affaire nouvelle signifie autoriser une équipe émettrice d'une fiche d'affaire nouvelle (FAN) à consacrer du temps pour élaborer une offre commerciale détaillée. Ceci entraîne bien évidemment un coût qui sera réintégré dans le contrat en cas de succès et imputé en pertes et profits en cas d'échec. Refuser l'affaire signifie ne pas engager le travail ni émettre l'offre commerciale concernant cette affaire. Une allocation budgétaire est généralement demandée pour chaque FAN. Elle est accordée entièrement ou partiellement. Cette situation illustre bien le dilemme du manager qui peut, par un choix plus ou moins judicieux, gagner ou perdre beaucoup d'argent (sur un budget annuel de l'ordre de 2 ou 3 M€, la pertinence des décisions joue sur 0.5 à 1 M€). Vue de l'équipe interne à l'entreprise et émettrice de FAN, la problématique est bien différente. Les collaborateurs sont le plus souvent en situation de juge et partie, cherchant généralement à obtenir par tous les moyens l'autorisation d'émettre l'offre et de collecter les budgets associés.

Le modèle d'aide à la décision retenu utilise la problématique du tri[2]. La réponse se fait en attribuant à chaque affaire nouvelle l'une des quatre classes suivantes :
– C1 : Non ;
– C2 : Dubitatif non ;
– C3 : Dubitatif oui ;
– C4 : Oui.

Les affaires nouvelles ne sont pas jugées les unes par rapport aux autres, mais évaluées indépendamment les unes des autres. L'association d'une catégorie à une affaire donne une indication dans l'absolu sur l'intérêt de l'affaire nouvelle. La problématique du tri est donc bien adaptée ici puisque les offres commerciales arrivent dans un ordre quelconque et indépendamment les unes des autres. Les problématiques du classement et de la sélection ne permettraient pas ici de donner de réponse utile, car la comparaison des offres serait impossible et inutile.

12.2.2 Les critères du modèle de première génération

Pendant de nombreuses années, les décisions de faire ou non des offres commerciales se prenaient localement de manière empirique et intuitive. Il était par ailleurs très difficile de consigner les données et résultats. En effet, en cas de succès, l'équipe était absorbée par les tâches nouvelles et, en cas d'échec, peu portée à faire de la publicité sur le sujet. Un comité "décideur" des affaires nouvelles fut ensuite créé, mais il contrôlait seulement 50 % des offres, le reste étant décidé localement sans en référer au comité. Une des conséquences fut d'ignorer longtemps le volume budgétaire exact de cette activité.

La validation du premier modèle a été faite en 1995, à partir d'un échantillon de onze offres commerciales typologiques caractérisant les quatre classes : "Oui, Oui dubitatif, Non dubitatif, Non".

Le modèle, mis en œuvre en 1996, était fondé sur neuf critères, répartis en trois familles, qualifiant les dimensions suivantes :
– probabilité de remporter l'affaire ;
– intérêt stratégique de l'affaire ;
– intérêt économique de l'affaire.

[2] La problématique du tri :
La problématique du tri consiste à affecter chacune des actions à une catégorie en examinant la valeur intrinsèque de l'action. Les catégories sont déterminées indépendamment des actions à affecter.
Rappel : Relation de surclassement (Roy, 1974) :
Une relation de surclassement est une relation binaire S définie dans l'ensemble A des actions, telle que, *aSb* si, étant donné la qualité des évaluations des actions et la nature du problème, il y a suffisamment d'arguments pour admettre que l'action *a* est au moins aussi bonne que l'action *b*, sans qu'il y ait de raison importante de refuser cette affirmation.

Famille n°1 : Probabilité de gagner l'appel d'offres

Critère 1 : Nature du contact avec le client.

Il correspond à la prise en compte des relations existantes, ou non, avec les clients qui émettent les appels d'offres. L'échelle de graduation était la suivante :

Mauvaise relation	Inconnu	Mal connu	Connu	Bonne relation	Excellente
0	2	3.5	6	7	8.5

Critère 2 : Compétence EADS LV et partenaires associés.

C'est la combinaison de la capacité technique de base par rapport aux concurrents pour réaliser le produit et de la capacité à l'appliquer dans le domaine considéré. L'échelle de graduation était la suivante :

Insuffisante	Fragile	Récente	Correcte	Assez bonne	Forte	Leader
0	2.5	5	7.5	11	12	16

Critère 3 : Compétitivité du prix EADS LV par rapport au prix du marché.

Pour avoir une chance de gagner l'appel d'offres, EADS LV doit vendre à un prix compétitif. L'échelle retenue était continue et variait de 0 à 2.

Famille n°2 : Intérêt stratégique

Critère 4 : Crédibilité de l'appel d'offres.

Ce critère doit permettre de détecter les appels d'offres douteux, pour lesquels EADS LV n'a aucune chance de l'emporter. L'échelle de graduation était la suivante :

Très peu crédible	Peu crédible	Douteux	Sans avis	Crédible	Très crédible
0	2	3.5	5.5	7.5	9.5

Critère 5 : Volonté stratégique.

Il s'agit de la volonté stratégique pour une affaire donnée, issue du plan stratégique de l'entreprise. Cette volonté est également en phase avec les investissements R&D. L'échelle de graduation était la suivante :

Exclu	Très faible	Faible	Moyen	Fort	Très fort	Obligatoire
0	6	7	9.5	10.5	11	14

Critère 6 : Disponibilité des équipes.

Ce critère prend en compte la planification de l'activité technique et commerciale d'EADS LV afin d'éviter les situations de crise dues aux surcharges momentanées. L'échelle de graduation était la suivante :

Indisponible	En suractivité	Très chargé	Chargé	Disponible	En sous-activité
0	3.5	6.5	10	12	14

Famille n°3 : Intérêt économique

Critère 7 : Effort de recherche autofinancée et d'investissement.

Ce critère prend en compte les éventuels besoins en cofinancement exigés par le client. L'échelle était continue et variait de 0 à 1.5 M€.

Critère 8 : Effort EADS LV.

Il s'agit du ratio de la dotation demandée sur le chiffre d'affaires HT prévu, destiné à renseigner et limiter les sommes mises en jeu pour financer la préparation de l'offre par rapport aux montants de gains espérés en cas de succès. L'échelle était continue et variait de 40 % à 0 %.

Critère 9 : Charges espérées en étude et en production.

Ce critère, qui représente les charges potentielles pouvant être générées en cas de succès, est destiné à favoriser le plein usage de l'outil de production. En conséquence, il favorise les grosses affaires. L'échelle variait de 0 à 530 000 heures.

12.2.3 Evolution du modèle de première génération

Pendant deux années, sachant que les appels d'offres n'étaient pas encore tous pris en compte, le comité a traité :

- en 1996, 104 offres commerciales ;
- en 1997, 96 offres commerciales.

Il est utile de rappeler qu'EADS LV peut se trouver dans deux situations pour un appel d'offres, soit en compétition, soit en gré à gré :

- en compétition, EADS répond à un appel d'offres ouvert à la concurrence ;
- en gré à gré, EADS répond à un appel d'offres en tant que client unique, ce qui ne signifie pas pour autant que l'affaire soit gagnée d'avance. En effet, dans ce dernier cas, l'entreprise doit se battre pour tenir dans le budget du client ou le prix du marché.

Au bout de deux ans de fonctionnement, une étude de performance fut conduite pour corréler les prédictions avec les résultats (succès ou échecs) :

- le taux d'erreur sur la prédiction des *oui* était de 65 % ;
- le taux d'erreur sur la prédiction des *non* était de 15 %.

A cette époque, les bons critères entrant dans la décision n'étaient pas forcément imaginables, et d'autres étaient difficilement réglables, car il fallait avoir du recul et de l'expérience pour les ajuster. Par exemple, un veto mûrement réfléchi sur le critère "Effort de recherche autofinancée" entraînait le refus des actions demandant un cofinancement de recherche supérieur à 500 k€, alors que le comité, en contradiction avec lui même, a retenu et gagné des actions demandant un cofinancement supérieur à 1 M€.

Pour clarifier cela, une étude de sensibilité sur les données cumulées 1996-1997 fut alors effectuée, en faisant varier uniquement les profils des actions de référence. Cinq scénarios ont été envisagés, parmi lesquels le scénario optimal retenu, qui devait nous conduire à un taux d'erreur de 15 % sur la prédiction des "oui" ayant une issue favorable (économie potentielle de 0.5 M€). A partir de janvier 1998, le nouveau réglage était opérationnel.

Dès 1999, le comité décideur voit ses pouvoirs considérablement renforcés. Il devient le comité des affaires nouvelles (CAN). Il garde un fonctionnement hebdomadaire et définit une nouvelle fiche d'affaire nouvelle (FAN) sur tableur Excel, pour mieux caractériser les actions candidates. Au demeurant, cette nouvelle FAN sert également de fiche de suivi tout au long de la préparation de l'offre et permet de corréler les hypothèses retenues au moment du choix avec les paramètres de l'offre finale envoyée au client. Le nouveau comité élimine

les dysfonctionnements en contrôlant strictement le budget de dotation (1.5 M€) et examine 90 % des offres courantes en 1999, puis 100 % en 2000. Les très petites affaires et les très grosses ne sont pas traitées par ce comité.

En 2000, une étude de performance étalée sur une durée de sept mois est entreprise. Un travail considérable est effectué sur les données afin de dégager un ensemble d'actions avec des indicateurs fiables. Le retour d'information sur les succès - échecs est laborieux. Il permet cependant de fournir des ratios précieux sur les performances commerciales de l'entreprise, qui sont présentés à la direction de l'entreprise.

Les méthodes et logiciels d'aide à la décision existent et peuvent être facilement mis en œuvre. Mais, après plus de cinq années d'expérience, la remise en cause du choix des critères par rapport à leur pertinence de sélectivité et la modélisation fine de ces critères apparaissent comme le point le plus essentiel dans l'aide à la décision. Au stade initial, sans données quantitatives historiques, la sélection des critères est normative, fortement dépendante de la culture des intervenants. Cela constitue donc un modèle de première génération, qui ne peut être qu'un exercice intellectuel purement cartésien, cherchant comment découper le problème en composantes cohérentes ou en critères.

En revanche, la pratique d'un modèle de première génération amène graduellement de la rigueur dans l'évaluation et une sélectivité des offres renforcée. Le gain de sélectivité constaté entre 1997 et 1998 (30 %) a engendré une économie de 500 k€ sur le budget de dotation. Le but à terme étant d'économiser bien plus.

12.3 Comprendre les insuffisances du modèle de première génération

De nombreux critères supposés importants se sont avérés non discriminants pour les raisons suivantes :

- les critères invérifiables autorisaient les notations “plaidoyer” et confinaient plus à l'enquête socio-culturelle qu'à la réalité de la compétition ;
- les acteurs par nature ne voulaient pas laisser échapper une affaire ;
- la routine induisait des décisions répétitives peu réfléchies.

Durant cette période, le périmètre et la nature de l'entreprise évoluèrent de manière considérable :

- vente de l'établissement Aérospatiale de Cannes (Satellites) à Alcatel en 1999 ;
- privatisation et entrée en bourse mi-1999 ;
- fusion Aérospatiale Matra mi-1999 ;
- fusion Aérospatiale Matra, DASA (Allemagne) et CASA (Espagne) en juillet 2001.

Ces modifications ont engendré des perturbations importantes dans la base de données des affaires et ont amené à abandonner les affaires 1996-1997 dans les échantillons de réglage du modèle.

Afin de mieux comprendre les insuffisances du modèle de première génération, l'étude suivante a révélé plusieurs types de dysfonctionnements :

- des critères qui s'avèrent non discriminants et donc ne servent jamais dans la décision ;
- des critères inopérants de par la méthode de notation et donc sans effet sur la décision. Ils doivent être corrigés ou supprimés ;
- des critères qui sont, en fait, des contraintes et sont donc toujours satisfaits.

12.3.1 Exemples de critères non discriminants

Nature du contact avec le client (critère 1)

En ce qui concerne la notation des affaires depuis 1996, le graphe suivant illustre la mauvaise répartition des scores enregistrés :

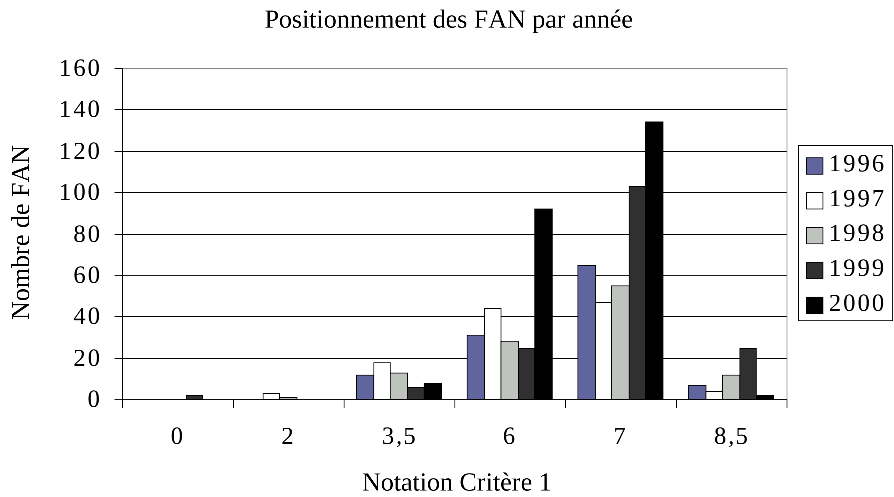


FIG. 12.1 – Nature du contact avec le client.

La relation avec le client semble être le plus souvent bonne (voir figure 12.1), la note la plus souvent attribuée étant la note “7”, mais ce n’est pas une raison suffisante, en temps de crise, pour que le client accorde un traitement privilégié aux offres commerciales EADS LV (voir figure 12.2).

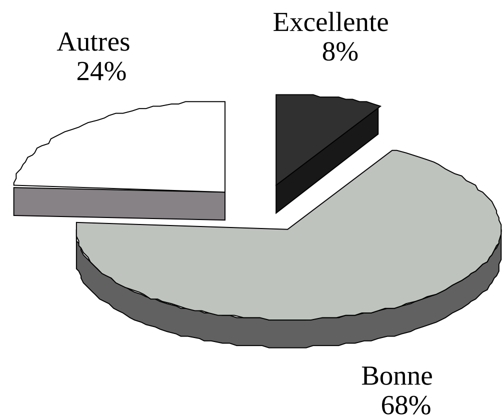


FIG. 12.2 – Nombre d’affaires perdues en fonction du type de relation avec les clients.

Crédibilité de l'appel d'offres (critère 4)

La situation pour le critère 4 "crédibilité de l'appel d'offres" est identique à celle du critère précédent. L'entreprise, dont le portefeuille de clients est très stable, n'a jamais rencontré d'appels d'offres douteux en six ans.

Disponibilité des équipes (critère 6)

Le plan de charge prévisionnel des équipes techniques et commerciales fournit, en théorie, l'information nécessaire à la notation. La distribution de la notation depuis 1996 sur ce critère, représentée à la figure 12.3, montre la difficulté de procéder à cette évaluation.

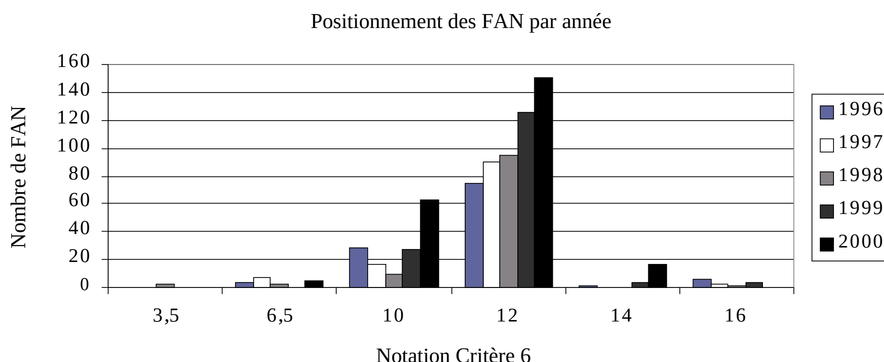


FIG. 12.3 – Fréquence des notes sur le critère 6.

Ce critère est remis en cause pour deux raisons. D'une part, il est difficile d'évaluer longtemps à l'avance le plan de charge des équipes techniques et commerciales et, d'autre part, il est très improbable qu'une affaire soit refusée sous prétexte de surcharge de travail des équipes. Enfin, s'il s'agit de la survie d'un secteur en crise, il est toujours possible d'obtenir une affaire en rendant une équipe disponible, soit en allongeant la durée de travail journalier, soit en travaillant les week-ends et jours fériés.

Charges espérées en étude et en production (critère 9)

L'analyse de la notation des affaires depuis 1996 sur ce critère représenté à la figure 12.4 montre que la majorité des affaires n'engendre que de faibles charges (inférieures à 50 000 heures).

Favoriser les "grosses" affaires demeure certes un élément de décision, mais n'a pas sa place dans le modèle, dans la mesure où cela reste un vœu pieux non suivi d'effet. Au demeurant, il est difficile de privilégier a priori une taille d'affaire idéale car, en période de crise, toute affaire est bonne à prendre.

12.3.2 Critères inopérants de par la méthode de notation

Compétence EADS LV et partenaires associés (critère 2)

Les intéressés ne peuvent évaluer eux-mêmes leur compétence, la compétition est la seule mesure objective, ce qui amène aux questions suivantes :
- Qui note le critère ?
- Qui est capable en interne de noter objectivement ?

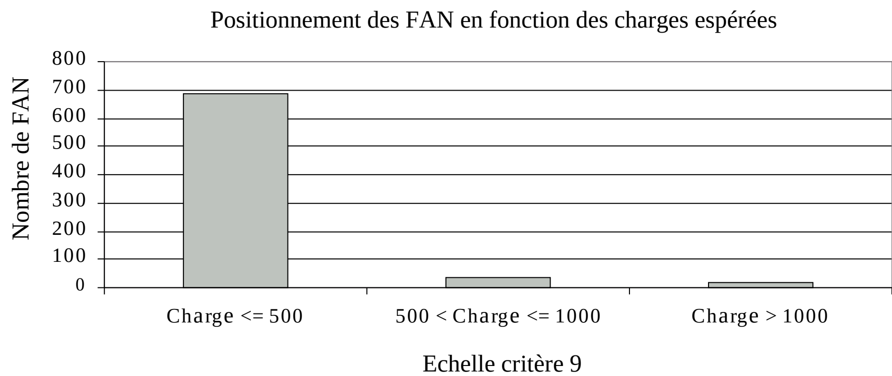


FIG. 12.4 – Fréquence des notes sur le critère 9.

– A-t-on intérêt à se faire noter par une entité externe à l'entreprise ?

Il est illusoire de vouloir mesurer un tel critère, issu d'une vision naïve et scolaire du problème, qui mène obligatoirement à des erreurs majeures de notation. Pour conserver un tel critère, il faut absolument résoudre le problème de notation.

Compétitivité du prix EADS LV par rapport au prix du marché (critère 3)

A cause de la complexité et du manque de transparence sur ce sujet, aussi bien en interne qu'en externe à l'entreprise, il s'avère extrêmement difficile de noter ce critère avec précision, le prix du marché étant le plus souvent difficile à appréhender. Dans l'impossibilité de trouver un ratio de notation qui soit un véritable instrument de calcul, ce critère a été abandonné au profit d'un critère technico-économique plus large, mais mesurable.

Volonté stratégique (critère 5)

Le plan stratégique d'EADS LV ne présentait pas, comme c'est le cas dans beaucoup de sociétés, d'éléments quantitatifs très détaillés sur les priorités de l'entreprise. Par ailleurs, les équipes techniques ou commerciales sont toujours persuadées de la haute priorité de leur activité (voir figure 12.5) et donc il ne faut surtout pas leur demander de noter elles-mêmes un tel critère. On remarquera incidemment que la volonté stratégique de l'entreprise n'a rien à voir, au premier degré, avec la probabilité de gagner ou non une affaire. Tout au plus, elle crée un terrain favorable et génère des investissements, qui ont un effet à moyen ou long terme.

12.3.3 Critères qui sont en fait des contraintes

Effort de recherche autofinancée et d'investissement (critère 7)

Une vingtaine d'affaires par année sont concernées par ce critère, l'effort de recherche autofinancée et d'investissement était échelonné entre 0 et 1.5 M€. Ce critère est représenté à la figure 12.6.

Ce critère n'est pas discriminant dans la mesure où, pour les affaires réellement stratégiques, une entreprise peut toujours se donner les moyens de répondre à la demande du client, notamment en termes d'investissements R&D. C'est en fait une contrainte qu'il faut prendre en compte et non pas un critère.

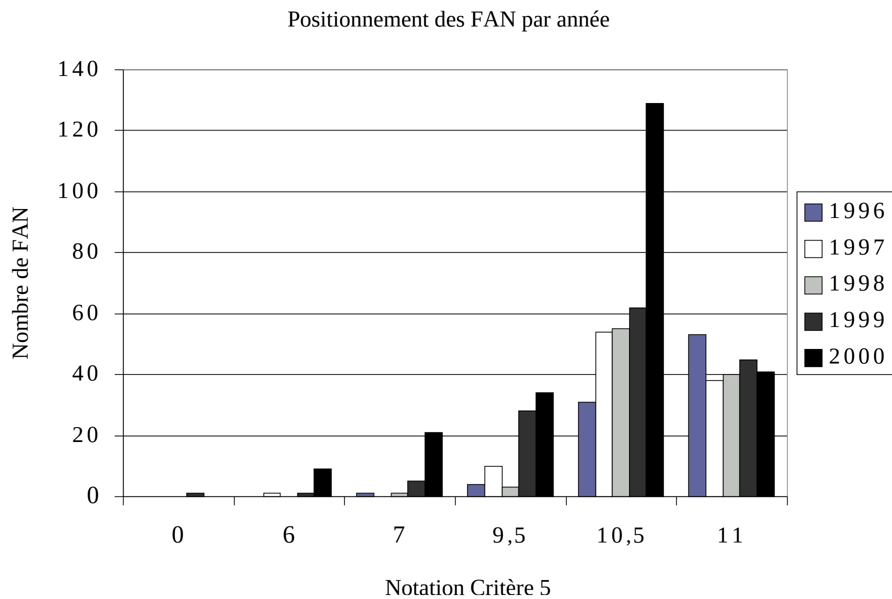


FIG. 12.5 – Volonté stratégique tenant compte de la R&D.

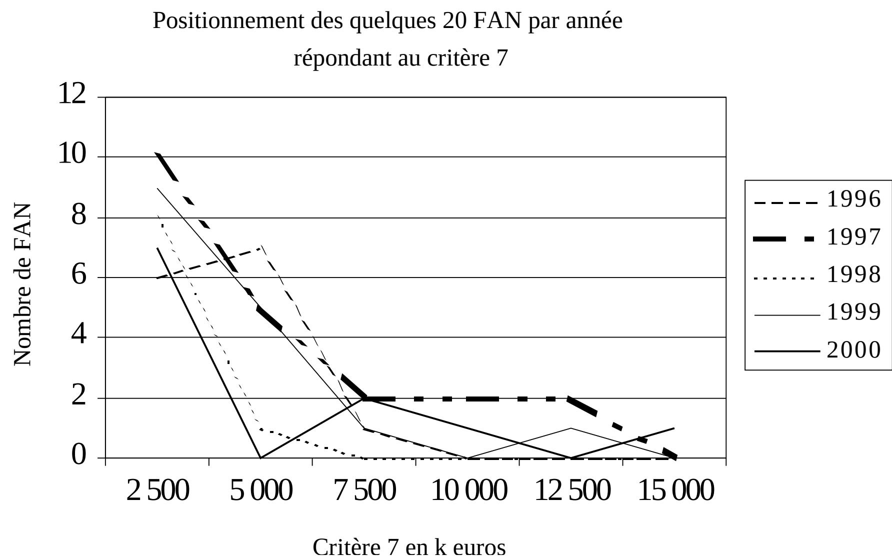


FIG. 12.6 – Effort de recherche autofinancée.

Effort EADS LV (critère 8)

Ce critère n'est pas discriminant, et se ramène à une contrainte qui maintient l'effort financier, affaire par affaire, dans une enveloppe raisonnable par rapport au chiffre d'affaires espéré. Comme le montrent les graphiques de la figure 12.7, cette règle est connue des acteurs en interne, elle est donc devenue une contrainte satisfaite par tous.

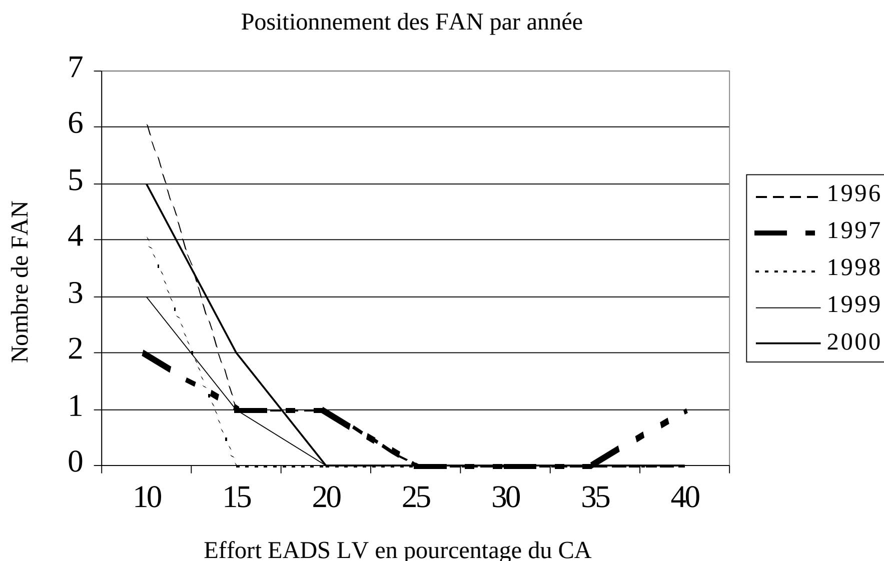


FIG. 12.7 – Fréquence des notes sur le critère 8.

12.4 Modèle de deuxième génération

L'évolution la plus fondamentale consiste en la modification de la question à laquelle le modèle d'aide à la décision de EADS LV tente de répondre. Le modèle ne tente plus de résoudre la question : "Faut-il ou non répondre à un appel d'offres ?" mais la question : "Va-t-on gagner ou non cet appel d'offres ?" En effet, cette modification essentielle se justifie par le fait que l'entreprise juge toujours les performances du modèle sur un résultat constaté (offre gagnée ou perdue).

La nouvelle problématique est plus restreinte que la précédente et les intérêts stratégiques et économiques n'y figurent plus. Ces intérêts sont bien sûr pris en compte par les membres du comité des affaires nouvelles (CAN), mais le modèle ne prétend plus répondre à toutes les questions en même temps. La question de savoir s'il faut répondre ou non appartient donc au comité, qui tient compte ou non des prédictions fournies, lors de sa réunion hebdomadaire.

12.4.1 Principe de reconstruction du modèle

Le nouveau modèle a été construit à partir des principes suivants :

- ne plus faire appel du tout à des experts en situation de juge et partie pour effectuer une notation ;
- utiliser, pour la notation, des critères basés sur des informations factuelles historiques cumulées, dont la précision augmente avec la quantité d'affaires traitées, donc au cours du temps ;
- remplacer les notions qualitatives par des notions quantitatives et éviter, autant que faire se peut, les critères purement qualitatifs ;
- utiliser l'expérience acquise depuis la création du modèle pour bâtir de nouvelles échelles. Toutes les caractéristiques des affaires sont consignées dans une base de données depuis 1996. La connaissance de la répartition des notes sur les années passées doit permettre de mieux définir les échelons, tant au niveau de leur position dans l'absolu que de leur éloignement relatif ;

– accéder facilement aux informations nécessaires à la notation, ce qui permettra, outre une notation plus performante, d'accélérer le processus de notation ;

En ce qui concerne les échelles et leurs échelons, les principes retenus sont :

– moins d'échelons, mais mieux répartis et faciles à distinguer entre eux ;
– lorsque l'on a recours malgré tout à des critères qualitatifs, il est indispensable de recourir à une matrice à deux dimensions, pour éviter le danger de la notation directe. Cela permet d'utiliser deux notions (une dans chaque dimension), ainsi la notation répond à deux questions précises, au lieu de conduire à une estimation intuitive hasardeuse.

12.4.2 Abandon du principe de modèle unique

Les affaires nouvelles étant très disparates, chaque affaire s'inscrit dans un environnement spécifique (situation commerciale, concurrence, client institutionnel ou industriel), concerne différents secteurs (missiles stratégiques, satellites, lanceurs, équipements, etc.) et des activités variées (fourniture d'équipements, prestation de services, études, etc.).

Il est donc apparu de manière évidente au cours des années d'utilisation du modèle que l'approche par un modèle unique et omnipotent était beaucoup trop rustique et restait limitée en capacité d'amélioration. Il a donc été décidé de créer des familles de modèles distincts, chacun d'eux ayant des réglages et/ou des critères particuliers, en fonction des activités de l'entreprise. A travers le plan commercial, une segmentation des activités de la société a été imposée avec succès dans l'entreprise et un modèle par segment a été réalisé pour les produits et pour les services. Les critères qualitatifs ont été remplacés par des critères quantitatifs basés sur des données factuelles accessibles, et mesurés par des scores enregistrés par l'entreprise, segment par segment.

Cette démarche a conduit à construire tout d'abord un modèle à quatre critères. Ce modèle n'expliquant pas la totalité du vaste échantillon utilisé, une étude approfondie, fondée sur les lois relatives aux familles cohérentes de critères de Bernard Roy, a permis d'identifier un $5^{\text{ième}}$ critère. Les échantillons de réglage du modèle étant constitués de la totalité des affaires disponibles pour un segment donné, il n'y a pas de limite dans la précision prévisible d'un tel modèle, qui pourra s'améliorer par augmentation du nombre de critères ou par affinage des ratios de mesure des critères.

12.4.3 Architecture des familles de modèles

Afin de simplifier l'utilisation des outils d'aide à la décision, un nombre limité de modèles a été retenu :

– le modèle **Electre Tri 2001 Produits**, conçu pour être appliqué aux FAN dont l'objet est la fourniture de produits ;
– le modèle **Electre Tri 2001 Services**, conçu pour être appliqué aux FAN dont l'objet est une prestation de services (études, essais, etc.).

Les cinq critères retenus pour le modèle ELECTRE 2001 Produits sont :

– critère 1 : Positionnement technico-économique du produit ;
– critère 2 : Maîtrise technologique et maturité du produit ;
– critère 3 : Positionnement des clients ;
– critère 4 : Crédibilité financière de l'affaire chez le client ;
– critère 5 : Positionnement concurrentiel.

Les positionnements sont évalués avec les nombres d'affaires perdues portés en abscisse et les nombres d'affaires gagnées en ordonnée. Les informations recueillies depuis 1996 permettent

d'effectuer des positionnements de clients, de produits, de segments. Un positionnement est relatif à un ensemble donné d'affaires, qui concernent :

- des clients : représentation de la performance commerciale de l'entreprise vis-à-vis de ses clients ;
- des segments : représentation de la performance commerciale d'un segment ;
- des produits : représentation de la performance commerciale d'un produit ou d'une famille de produits.

La note étant calculée à partir de la position d'un client, d'un segment ou d'un produit, ces positionnements permettent de fonder l'évaluation quantitative sur des faits objectifs, ce qui est l'une des contraintes imposées au nouveau modèle.

12.4.4 Critères

Positionnement technico-économique du produit

Ce critère a été établi sur des données factuelles, permettant de rendre compte de la compétitivité technico-économique du produit ou de l'historique de la relation avec le client. Il est fondé sur les affaires nouvelles dont le statut définitif (gagné ou perdu) est connu (affaires recensées depuis 1996).

Chaque produit, famille de produits ou chaque client est positionné sur le graphe de la figure 12.8, appelé positionnement produit ou client, où le nombre d'affaires nouvelles perdues est porté en abscisse et le nombre d'affaires nouvelles gagnées en ordonnée. La surface du disque associé à un produit ou à un client est proportionnelle au chiffre d'affaires gagné avec le produit ou le client.

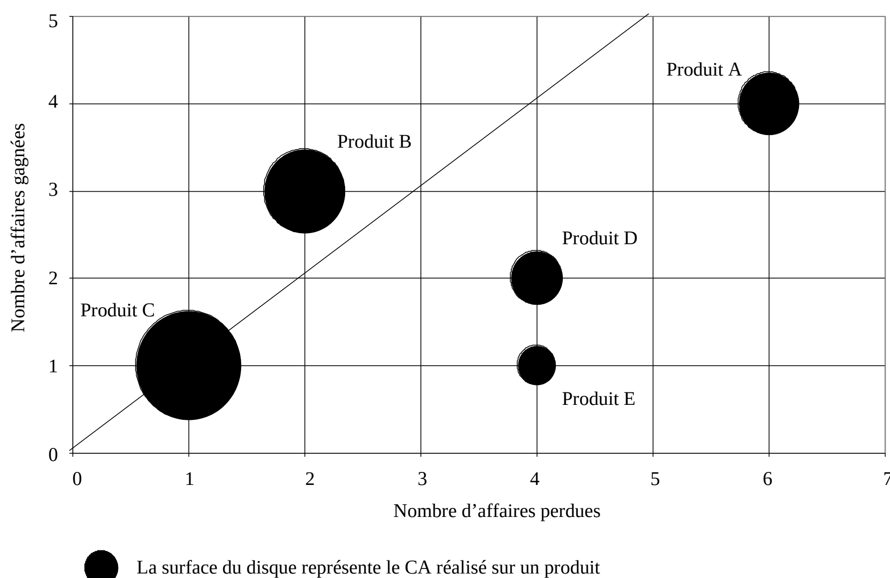


FIG. 12.8 – Positionnement produit (allure).

Maîtrise technologique et maturité du produit

Ce critère permet d'évaluer le niveau de développement du produit et son avance technologique, et d'en déduire sa compétitivité technique. Une nomenclature des produits a été

établie et imposée dans l'entreprise, afin de caractériser chaque produit, qui est ainsi positionné sur sa courbe de vie. Chaque position correspond à une étape du développement du produit ou de sa vie commerciale. En matière spatiale, les séries sont faibles et souvent réalisées sur mesure, ainsi la qualification en vol est une étape très importante pour les produits. La note est établie en fonction de la position du produit sur la courbe de la figure 12.9.

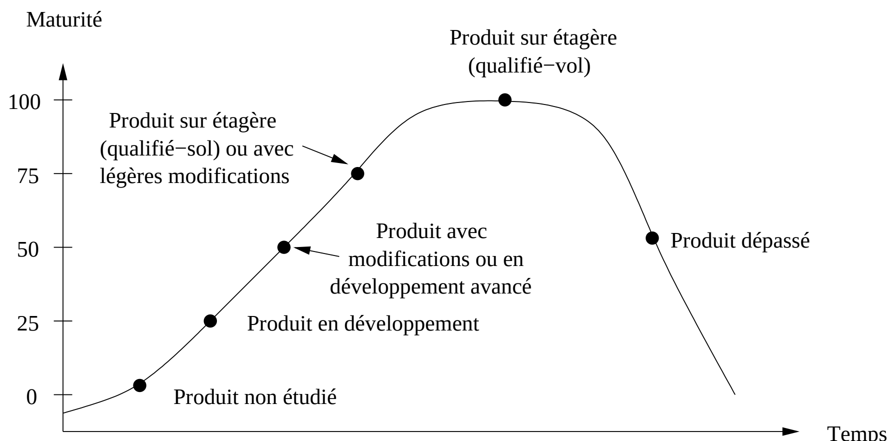


FIG. 12.9 – Courbe de vie des produits (allure).

Positionnement des clients

Ce critère est construit de façon analogue au critère positionnement technico-économique du produit.

Crédibilité financière de l'affaire chez le client

Ce critère prend en compte, autant que faire se peut, les risques d'annulation d'un projet. EADS LV n'ayant pas en général d'influence sur les risques d'annulation, le critère choisi est la crédibilité financière de l'affaire chez le client, qui rend compte de la crédibilité générale du projet.

Aussi la crédibilité de l'affaire dépend de deux facteurs financiers qui servent de dimension à la matrice de notation.

C'est lors de l'étude des affaires relatives au segment Réservoirs haute pression qu'il a été constaté que la moitié des affaires perdues étaient des affaires classées sans suite, les clients n'ayant pas mené à bien les projets. Ce fut le cas en particulier pour les constellations de satellites, projets ambitieux qui ont tous été plus ou moins abandonnés. Il est donc primordial d'avoir un critère permettant d'évaluer cette crédibilité. Les causes d'annulation sont nombreuses et l'évaluation de la situation difficile.

Nous distinguons :

– La crédibilité du client : les affaires classées “sans suite” sont déjà prises en compte dans le critère positionnement client.
– La probabilité d'intervention d'un événement extérieur : une décision politique ou une redistribution budgétaire peuvent remettre en cause une affaire. Ces événements sont par nature extrêmement difficiles à prévoir, mais les indices jalonnant les projets sont identifiés.

– Les difficultés techniques insurmontables : un projet ambitieux peut être annulé, car irréalisable du point de vue technique, compte tenu de l'enveloppe budgétaire.
– La prise en compte du niveau de sous-traitance : EADS LV peut être sous-traitant d'une entreprise maître d'œuvre. Au moment où la FAN est émise, le marché n'est pas gagné. Si le maître d'œuvre perd le marché, EADS LV, en tant que sous-traitant, perd aussi l'affaire.
– L'affaire prise très tôt : l'expérience a cependant montré que la probabilité d'annulation d'un projet n'évolue pas significativement dans le temps. Un projet a autant de chances d'être annulé à ses débuts qu'en phase avancée.
– Le contexte de l'affaire peut donner une indication sur la crédibilité de l'affaire. Que faire si le contexte est nouveau ?

Positionnement concurrentiel

Ce critère est un critère d'ajustement, qui permet de modifier la sévérité du modèle, et donc de façonner ses performances de prédiction du modèle, pour coller à la réalité commerciale de deux types de situations : la compétition et le gré à gré, qui ont des taux de succès très différents. Le principe retenu pour la notation est un principe matriciel calé sur les valeurs quantitatives récentes mesurées, et qui entrent dans le tableau de bord des performances commerciales de la société.

12.4.5 Paramétrage du modèle

En cela la méthode s'avère précieuse et bien adaptée.

Les modèles de première et seconde génération utilisent la même méthode Electre Tri, développée par le LAMSADE de l'université Paris-Dauphine, et son logiciel associé pour traiter les notes issues des critères. La présentation de la méthode, supposée connue, ne fait pas l'objet de ce chapitre. La problématique du tri, qui consiste à affecter chacune des actions à une catégorie en examinant la valeur intrinsèque de l'action, permet de traiter une seule action candidate ou un seul appel d'offres à la fois. En cela, la méthode est précieuse et bien adaptée à la réalité.

Les méthodes d'optimisation conduisent souvent à utiliser une fonction d'utilité (ou critère unique de synthèse) qui permet de classer les actions de la meilleure à la moins bonne. Ces méthodes reposent sur des hypothèses fortes et il vaut peut-être mieux se contenter d'un résultat moins riche en évitant l'introduction d'hypothèses mathématiques fortes et l'obligation de poser au décideur des questions trop difficiles. Pour ce faire, le concept de surclassement (relation emboîtée de concordance et de non discordance) proposé par Bernard Roy a été adapté à la problématique du tri. La transformation de la relation de surclassement valuée en une relation de surclassement nette S a été réalisée en utilisant un niveau de coupe l de 0.75. La procédure pessimiste, qui consiste à affecter l'action candidate ***a*** à la catégorie ***Ci*** la plus haute telle que l'action ***a*** surclasse l'action de référence basse ***bi*** (profiles) de la catégorie a été retenue. La totalité des capacités d'Electre Tri n'a pas été utilisée, les seuils de préférence ***p*** et d'indifférence ***q*** sont égaux, et en particulier le seuil de veto ***v*** n'a pas été utilisé.

12.4.6 Performance du modèle

L'appréciation des performances de prédiction, face à la réalité des affaires gagnées ou perdues, est assez complexe, puisqu'il convient de raisonner et de travailler par segment et

pour les quatre types de prédictions, à savoir les quatre classes : “Oui, Oui dubitatif, Non dubitatif, Non”.

Pour faire simple, la précision actuelle est remarquable, à l’aide du modèle Electre Tri 2001 à cinq critères, le calcul des scores gagnés - perdus est reconstitué avec une précision de 96 % à 100 % pour les catégories C1 (Non) et C4 (Oui).

Le score obtenu sur le segment fluidique présenté à titre d’exemple à la figure 12.10 montre une précision de 100 % pour les catégories C1 (Non) et C2 (Non dubitatif), un taux d’erreur de 12.5 % sur la catégorie C3 (Oui dubitatif) et une précision de 100 % pour la catégorie C4 (Oui).

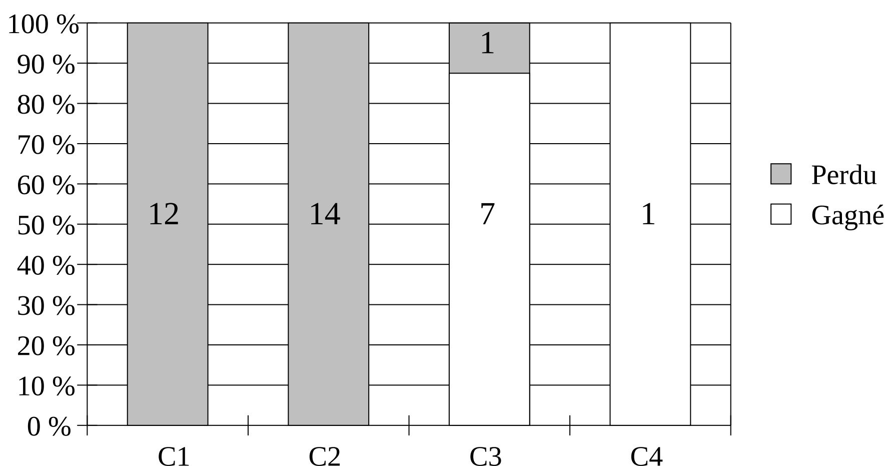


FIG. 12.10 – Segment fluidique : résultat sur échantillon.

12.5 Conclusion

La maîtrise des données commerciales nécessaires pour mettre en œuvre le modèle a permis la diffusion mensuelle d’un tableau de bord commercial, contribuant ainsi à un meilleur pilotage de l’entreprise.

Ce tableau de bord se compose actuellement de :

– statistiques consolidées et par segment sur les affaires gagnées - perdues ;
– dix-sept positionnements de produits ou de clients ;
– un certain nombre de ratios pertinents.

Ce tableau de bord est représenté à la figure 12.11.

Le modèle a eu un impact important sur l’élaboration du plan stratégique de la société, en introduisant la liste des priorités pour tous les produits et les services. Il a également contribué à imposer une segmentation des activités, qui est devenue l’ossature du plan commercial, et il a poussé à la restructuration de la nomenclature des produits.

Les perspectives d’évolution du modèle sont tournées vers une homogénéité de la précision dans tous les segments, vers une éventuelle modification des classes d’équivalence, vers la subdivision des problèmes par une modélisation de plus en plus fine et vers une décentralisation des outils d’aide à la décision à l’intérieur de l’entreprise. A l’instar des systèmes experts, la plupart des ratios servant à la notation des critères cumulent l’expérience. Ainsi toute situation nouvelle, après une phase d’erreur de prédiction, sera automatiquement intégrée dans les prédictions suivantes. Enfin, pour les produits récurrents, il est possible de

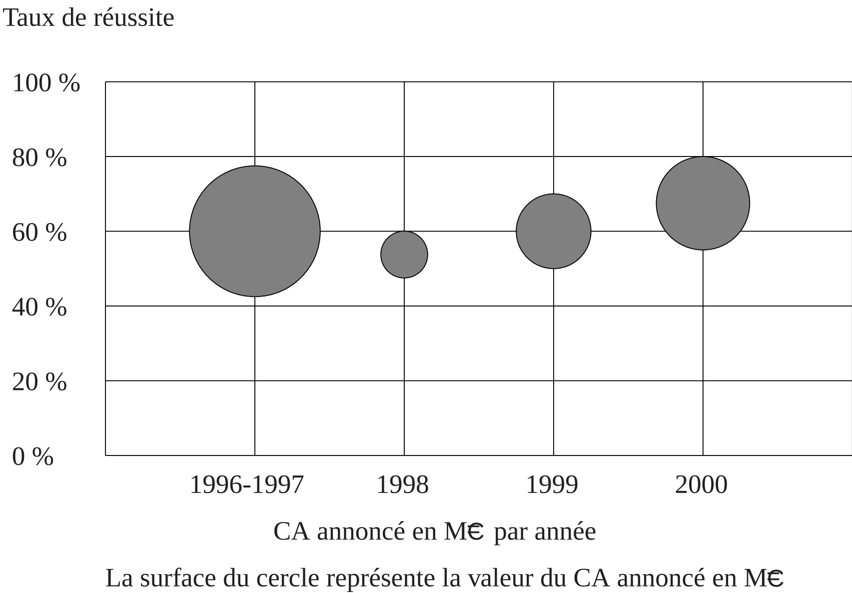


FIG. 12.11 – Exemple de ratio, taux de réussite commercial, CA annoncé en M€.

reconstituer le passé et de prévoir l'avenir avec une très grande précision dans les cas d'évolution linéaire.

Compte tenu des rôles fédérateur et planificateur des outils d'aide à la décision, qui amènent l'entreprise à se doter de bases de données quantitatives, favorables au pilotage de l'entreprise, la généralisation de ces outils à tous les problèmes d'allocation budgétaire est prévisible à moyen terme.

L'aide à la décision pour ce problème d'allocation budgétaire dans les offres commerciales est dotée d'un avenir prometteur, car les économies potentielles sont considérables et ne concernent pas uniquement le monde spatial. En particulier, compte tenu des abus constatés ces dernières années, la tendance est nettement à la généralisation des appels d'offres pour l'attribution de marchés, notamment en ce qui concerne les collectivités.

Conclusion

Comme nous avons pu le voir tout au long de cet ouvrage, l'optimisation multiobjectif n'est pas une tâche facile.

Tout commence par la définition d'un optimum multiobjectif. Une définition générale et bien établie est celle de Pareto. Mais cette définition étant trop générale, des variantes ont été introduites : optimalité lexicographique, optimalité maximale, etc.

Ensuite vient le mode de résolution. En fonction de la connaissance que l'on a du problème, on peut :

– partir à la recherche d'une unique solution si les préférences du décideur sont bien déterminées et si celui-ci ne souhaite pas choisir parmi un éventail de possibilités ;
– préciser les préférences du décideur en cours d'optimisation ;
– approcher la surface de compromis par un "nuage" de solutions.

On a aussi une grande latitude quant au choix du traitement du problème multiobjectif. On peut se ramener à une seule et unique fonction objectif ou alors traiter vectoriellement le problème.

Pour finir, il faut choisir une méthode d'optimisation pour résoudre le problème multiobjectif. Cette dernière tâche est déjà délicate en optimisation monobjectif, alors, on conçoît que l'optimisation multiobjectif soit d'une difficulté plus grande encore.

Cependant, le jeu en vaut la chandelle. L'optimisation multiobjectif offre à l'utilisateur davantage de degrés de liberté pour modéliser son problème. Elle souligne aussi certaines difficultés de modélisation que l'on élude souvent en optimisation monobjectif. En effet, avant de résoudre un problème d'optimisation monobjectif, on fait souvent un compromis entre certains objectifs, à l'insu du concepteur du modèle. L'optimisation multiobjectif porte au grand jour ce compromis et c'est aussi cela la difficulté de l'optimisation multiobjectif : une gène occasionnée par un compromis difficile à réaliser. Il faut bien garder à l'esprit que l'optimisation multiobjectif ne traitera pas d'un coup de baguette magique ce compromis. Ce sera toujours au décideur de trancher et de choisir une solution plutôt qu'une autre. L'optimisation multiobjectif offre, cependant, des outils permettant d'aider le décideur à effectuer son choix.

Plusieurs domaines de l'optimisation multiobjectif font l'objet, actuellement, d'une attention particulière :

– **Le contrôle de la diversité des solutions sur la surface de compromis :**
  Dans la méthode de résolution *a posteriori*, on cherche à obtenir une approximation de la surface de compromis. Une bonne approximation correspond à un nuage de points uniformément répartis sur celle-ci. Plusieurs méthodes (dont certaines en cours de développement) s'efforcent d'atteindre ce but.
  On peut aussi chercher à obtenir une bonne diversité des solutions dans l'espace des paramètres.

– **L'élaboration de méthodes d'optimisation multiobjectif permettant d'obtenir une approximation de la surface de compromis plus rapidement** :
  Il y a beaucoup de degrés de liberté sur lesquels on peut jouer pour améliorer la vitesse de convergence d'une méthode d'optimisation multiobjectif. Par exemple, on peut :
  - utiliser une métaheuristique différente (recuit simulé, algorithmes génétiques, etc.) ;
  - utiliser une méthode de traitement multiobjectif différente ;
  - utiliser un voisinage plus adapté au problème que l'on traite ;
  - etc.

– **La mise au point de méthodes permettant d'intégrer les préférences d'un utilisateur pour une résolution *a priori*** :
  Depuis quelques années, on voit paraître des articles traitant de la résolution de problèmes multiobjectifs de manière *a priori*, à l'aide des outils de la théorie des préférences ou de l'aide à la décision. On peut penser qu'une jonction entre l'optimisation multiobjectif et l'aide à la décision va s'opérer prochainement.

– **L'étude de méthodes de visualisation de données permettant de mieux appréhender les résultats d'une optimisation multiobjectif** :
  Il existe encore très peu de méthodes de visualisation de données dédiées à la représentation de résultats d'optimisation multiobjectif. Pourtant, dans le domaine des statistiques, de nombreuses méthodes sont disponibles, par exemple :
  - l'analyse en composantes principales ;
  - la projection non linéaire ;
  - etc.

– **L'élaboration de nouvelles métriques permettant de juger, en cours d'optimisation, de la qualité des résultats intermédiaires** :
  Certaines métriques, bien que très utiles, ont un domaine d'application restreint. D'autres nécessitent des informations qu'il est impossible d'obtenir avec un problème "réel", ce qui restreint leur emploi à des problèmes de test.
  Certaines techniques du domaine de l'économie sont applicables à l'optimisation multiobjectif, comme l'analyse d'enveloppe de données.

L'optimisation multiobjectif est donc appelée à se développer dans le futur comme en témoignent le nombre sans cesse croissant de publications et la création de nouvelles conférences consacrées à ce sujet.

Ce développement peut être aussi suivi à travers l'arrivée des idées de l'optimisation multiobjectif dans les entreprises : celles-ci sont, de plus en plus, confrontées à la quadrature du cercle d'un compromis nécessaire entre le coût, la sécurité et la qualité.

Bibliographie

[Alexandrov et al. 94] N. Alexandrov, J. E. Dennis, *Algorithms for bilevel optimization*, Rapport Technique TR-94-77, septembre 1994.

[Altenberg 97] L. Altenberg, *Fitness Distance Correlation Analysis : An Instructive Counterexample*, Proc. of the Seventh Int. Conf. on GA (ICGA97), San Francisco, CA, 1997.

[Altschuler et al. 94] E. L. Altschuler, T. J. Williams, E. R. Ratner, F. Dowla, F. Wooter, *Method of constrained global optimization*, Physical review letters, janvier 1994.

[Andreatta et al. 97] A. A. Andreatta, S. E. R. Carvalho, C. C. Ribeiro, *A Framework for the Development of Local Search Heuristics for Combinatorial Optimization Problems*, PhD report, Catholic University of Rio de Janeiro, 1997.

[Andreatta et al. 98] A. A. Andreatta, S. E. R. Carvalho, C. C. Ribeiro, *An Object Oriented Framework for Local Search Heuristics*, Rapport Technique, University of Rio de Janeiro, 1998.

[Argaud 92] J. P. Argaud, *Une méthode d'optimisation des plans de rechargement pour la gestion du combustible nucléaire*, Note interne EDF-DER numéro HI-072/96/013/0, janvier 1992.

[Ascend] Logiciel Ascend 10.1

[Asselmeyer 96] T. Asselmeyer, W. Ebeling, *Unified Description of Evolutionary Strategies over Continuous Parameter Spaces*, mai 1996.

[Ausiello et al. 99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protrasi, *Complexity and Approximation Combinatorial Optimization Problems and their Approximability Properties*, Springer Verlag, 1999. (document), 5.8

[Azarm 96] S. Azarm, *Multiobjective optimum design*, 1996. 2.3.1, 2.7.3, 2.13

[Azarm et al. 99] S. Azarm, B. J. Reynolds, S. Narayanan, *Comparison of two Multiobjective Optimization Techniques with and within Genetic Algorithms*, Proc. of the 1999 ASME Design Engineering Technical Conference, septembre 1999.

[Baldick et al.] R. Baldick, A. B. Kahng, A. Kennings, I. L. Markov, *Efficient Optimization by Modifying the Objective Function : Applications to Timing-Driven VLSI Layout*.

[Barda 87] O. H. Barda, *Etude comparative des méthodes multicritères dans le cadre d'un problème de localisation*, Université Paris-Dauphine, 1987.

[Barnett 97] L. Barnett, *TANGLED WEB : Evolutionary Dynamics on Fitness Landscapes with Neutrality* MSc Dissertation, School of Cognitive Sciences, University of East Sussex, Brighton, 1997.

[Bartak 99] R. Bartàk, *Constraint programming : in pursuit of the holy grail*, Proceedings of WDS99, Prague, juin 1999.

[Battiti 94] R. Battiti, G. Tecchiolli, *The reactive tabu search*, ORSA Journal on computing, volume 6, numéro 2, pages 126-140, 1994.

[Ben-Tal et al. 96] A. Ben-Tal, J. Tind, *Duality with Multiple Criteria and Multiple Resources*, 1996.

[Boese et al. 94] K. D. Boese, A. B. Kahng, *Best-So-Far vs. Where-You-Are : Implications for Optimal Finite-Time Annealing*, Systems and Control Letters, volume 22, numéro 1, pages 71-80, janvier 1994.

[Boese et al.] K. D. Boese, A. B. Kahng, S. Muddu, *A New Adaptative Multi-Start Technique for Combinatorial Global Optimizations*, Op. Res. Letters, volume 16, numéro 2, septembre 1994, pages 101-113.

[Bonomi 88] E. Bonomi, J.-L. Lutton, *Le recuit simulé*, Pour la Science, numéro 129, pages 68-77, juillet 1988. 5.4

[Borges et al. 00] C. C. H. Borges, H. J. C. Barbosa, *A Non-generational Genetic Algorithm for Multiobjective Optimization*, 2000 Congress on Evolutionary Computation, volume 1, pages 172-179, juillet 2000.

[Bourles] H. Bourles, *Optimisation et commande optimale*, Notes de cours, E.N.S. Cachan, 1997.

[Bowlin 98] W. F. Bowlin, *Measuring Performance : An Introduction to Data Envelopment Analysis (DEA)*, Journal of Cost Analysis, pages 3-27, 1998.

[Boyan et al. 97] J. A. Boyan, A. W. Moore, *Using Prediction to Improve Combinatorial Optimization Search*, Sixth International Workshop on Artificial Intelligence and Statistics (AISTATS), 1997.

[Brans et al. 86] J. P. Brans, Ph. Vincke, B. Mareschal, *How to select and how to rank projects : The PROMETHEE method*, European journal of Op. Research, volume 24, pages 228-238, 1986. 6.3.8.1, 6.3.9.1

[Bruyere et al. 86] M. Bruyère, A. Vallée, Ch. Collette, *Extended fuel cycle length*, Note Framatome, FRADOC 11, septembre 1986.

[Burke et al. 92] L. I. Burke, J. P. Ignizio, *Neural networks and operations research : an overview*, Computer Operations Research, volume 19, numéro 3/4, pages 179-189, 1992.

[Cardon et al. 00] A. Cardon, T. Galinho, J.-P. Vacher, *Genetic algorithms using multi-objectives in a multi-agent system*, Robotics and Autonomous Systems, volume 33, pages 179-190, 2000.

[Clerc 00] M. Clerc, *Discrete Particle Swarm Optimization Illustrated by the Travelling Salesman Problem*, février 2000.

[Charon et al. 96] I. Charon, A. Germa, O. Hudry, *Méthodes d'optimisation combinatoire*, éditions Masson, 1996.

[Chekroun 85] E. Chekroun, *Aide à la décision*, éditions Sirey, mars 1985.

[Coello 96] C. A. Coello Coello, *An empirical study of evolutionary techniques for objective optimization in engineering design*, PhD Thesis, Department of Computer Science, Tulane university, New Orleans, LA, avril 1996.

[Coello 98] C. A. Coello Coello, *An updated survey of G.A. based multiobjective optimization techniques*, Rapport technique Lania-RD-98-08, Xalapa, Veracruz, Mexico, décembre 1998. 2.1.1, 2.7.3, 2.8.1, 5.7.1.1.1

[Coello 99] C. A. Coello Coello, *A survey of constraint handling techniques used with evolutionary computation methods*, Rapport technique Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, Mexico, 1999.

[Coello et al. 98] C. A. Coello Coello, A. D. Christiansen, *Two new GA-based Methods for Multiobjective optimization*, Civil Engineering Systems, volume 15, numéro 3, pages 207-243, 1998.

[Coello et al. 99] C. A. Coello Coello, A. D. Christiansen, *MOSES : A Multiobjective Optimization Tool for Engineering Design*, Engineering Optimization, volume 31, numéro 3, pages 337-368, 1999. 1.5

[Coello et al. 00] C. A. Coello Coello, *Handling Preferences in Evolutionary Multiobjective Optimization : A Survey*, 2000 Congress on Evolutionary Computation, volume 1, pages 30-37, juillet 2000.

[Coello 01] C. A. Coello Coello, *A Short Tutorial on Evolutionary Multiobjective Optimization*, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 21-40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[Coleman et al. 93] T. Coleman, D. Shalloway, Z. Wu, *Isotropic Effective Energy Simulated Annealing Searches for Low Energy Molecular Cluster States*, Comp. Opt. and App., numéro 2 pages 145-170, 1993.

[Collette et al. 00] Y. Collette, P. Siarry, H.-I. Wong, *A systematic comparison of performance of various multiple objective metaheuristics using a common set of analytical test functions*, Foundations of Computing and Decision Sciences, Volume 25, numéro 4, pages 249-272, 2000.

[Collins 96] T. Collins, *Genotype-Space Mapping : Population Visualization for Genetic Algorithms*, Rapport Technique KMI-TR-39, Knowledge Media Institute, Septembre 1996.

[Cooper et al. 00] W. W. Cooper, L. M. Seiford, K. Tone, *Data Envelopment Analysis : A comprehensive Text with Models, Applications, References and DEA-Solver Software*, Kluwer Academic Publishers, 2000. 7.16

[Corne et al. 00] D. W. Corne, J. D. Knowles, M. J. Oates, *The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization*, In 6th International Conference on Parallel Problem Solving from Nature PPSN VI, pages 839-848, septembre 2000, Paris.

[C.S.E.P. 95] The computer science education project, *Mathematical optimization*, Electronic book, 1995.

[Cvetkovic et al. 99] D. Cvetkovic, I. C. Parma, *Use of preference G.A.-Based multi-objective optimization*, Proceedings of the Genetic and Evolutionary Conference : GECCO99, volume 2, pages 1504-1509, Orlando, Floride, USA, juillet 1999.

[Cvetkovic 00] D. Cvetkovic, *Evolutionary Multi-Objective Decision Support Systems for Conceptual Design*, PhD, School of Computing, Faculty of Technology, University of Plymouth, November 2000.

[Daniels 78] R. W. Daniels, *An introduction to numerical methods and optimization techniques*, éditions North Holland New-York, 1978.

[Darwen et al. 95] P. Darwen, X. Yao, *A dilemma for fitness sharing with a scaling function*, Proceedings of the second IEEE International Conference on Evolutionary Computation, Piscataway, New Jersey, pages 166-171, 1995.

[Das et al. 98] I. Das, J. Dennis, *Normal boundary intersection : a new method for generating Pareto optimal points in multicriteria optimization problems*, SIAM J. on Optimization, volume 8, numéro 3, pages 631-657, août 1998.

[Das 99] I. Das, *A Preference Ordering Among Various Pareto Optimal Alternatives*, Structural Optimization, volume 18, number 1, pages 30-35, august 1999.

[Dauer et al. 80] J. P. Dauer, R. J. Krueger, *A multiobjective optimization model for water resources planning*, Applied Mathematical Modelling, pages 171-175, juin 1980.

[Davis 91] L. Davis, *Handbook of Genetic Algorithms*, éditions Van Nostrand Reinhold, 1991. 5.8

[Deb 98a] K. Deb, *Multiobjective genetic algorithms : problem difficulties and construction of test problems*, Technical Report CI-49/98, Dortmund, Department of Computer Science/LS11, University of Dortmund, Allemagne, 1998. 8.2, 8.4

[Deb 98b] K. Deb, *Non-linear goal programming using multi-objective genetic algorithms*, Rapport technique CI-60/98, Dortmund, Department of Computer science/LS11,University of Dortmund, Allemagne, octobre 1998.

[Deb 99] K. Deb, *An overview of multi-objective evolutionary algorithms*, Journée J.E.T., Copie de transparents, mai 1999. 1.6.2, 5.7.1.1.1, 5.7.2.1.1

[Deb et al. 00a] K. Deb, T. Goyal, *Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence*, Rapport Technique, KanGAL report 200004, Indian Institute of Technology, Kanpur, Inde, 2000.

[Deb et al. 00b] K. Deb, A. Pratap, T. Meyarivan, *Constrained Test Problems for Multi-Objective Evolutionary Optimization*, Rapport Technique, KanGAL 200005, Institute of Technology of India, Kanpur, Inde, 2000.

[Deb et al. 00c] K. Deb, E. Zitzler, L. Thiele, *Comparison of Multiobjective Evolutionary Algorithms : Empirical Results*, Technical Report TIK, 2000. 11.4.2.5

[Deb 01] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001. 1.8.5, 5.7.2.3.3, 5.8

[Deb et al. 01a] K. Deb, T. Goel, *A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design*, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 385-399. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[Deb et al. 01b] K. Deb, A. Pratap, T. Meyarivan, *Constrained Test Problem for Multi-Objective Evolutionary Optimization*, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 284-298. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[Deb et al. 01c] K. Deb, T. Goel, *Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence*, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 67-81. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[Deb et al. 01d] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II*, In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo and Hans-Paul Schwefel (editors), Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 849-858. Springer, 2000.

[Delbos 75] M. Delbos, *Le positionnement multidimensionnel*, Note interne EDF-DER numéro HI/1943/02, septembre 1975.

[Deng et al. 96] H. L. Deng, W. Gouveia, J. Saales, *The CWP object-oriented optimization library*, 1996.

[Diamantaras et al. 99] K. I. Diamantaras, K. Hornik, M. G. Strintzis, *Optimal Linear Compression Under Unreliable Representation and Robust PCA Neural Models*, IEEE Trans. on Neural Networks, volume 10, numéro 5, pages 1186-1195, 1999.

[Di Gaspero et al. 00] L. Di Gaspero, A. Schaerf, *EasyLOCAL++ : An object oriented framework for flexible design of local search algorithms*, Rapport Technique UDMI/13/2000/RR, 2000.

[Dorigo et al. 96] M. Dorigo, V. Mariezzo, A. Colorni, *The Ant system : optimization by a colony of cooperating agents*, IEEE Trans. on Systems, Man, and Cybernetics-part B, volume 26, numéro 1, pages 1-13, 1996.

[Doye et al.] J. P. K. Doye, D. J. Wales, *On the Thermodynamics of Global Optimization*, Phys. Rev. Letter, volume 8, pages 1357-1360, 1998.

[Dozier et al. 98] G. Dozier, S. McCullough, A. Homaifar, L. Moore, *Multi-objective evolutionary path planning via fuzzy tournament selection*, IEEE Internat. Conf. on Evolutionary Comp., pages 684-689, Piscataway, New Jersey, mai 1998.

[Dreyfus et al. 01] G. Dreyfus, J.-M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, *Réseaux de neurones : méthodologie et applications*, Eyrolles, 2001. 10.3

[Duch et al. 98] W. Duch, J. Korczak, *Optimization and Global minimization methods suitable for neural networks*, Neural Computing Survey (submitted), 1998.

[Duch 99] W. Duch, *Alternative to Gradient Based Neural Training*, In Fourth Conf. on Neural Networks and their Applications, Zakopane, Poland, mai 1999.

[Duvivier et al. 96] D. Duvivier, Ph. Preux, E.-G. Talbi, *Climbing up NP Hard Hills*, Fourth Int. Conf. on Parallel Problem Solving from Nature, septembre 1996.

[Edelman 97] S. Edelman, N. Intrator, *Learning as Extraction of Low-dimensional representations*, Mechanisms of Perceptual Learning, Psychology of Learning and Motivation series, Academic Press, Berlin, Allemagne, pages 574-583, 1997.

[Ehrgott 97] M. Ehrgott, *A characterization of Lexicographic Max-ordering Solutions*, Methods of Multicriteria Decision Theory : Proceedings of the 6th Workshop of the DGOR Working Group Multicriteria and Decision Theory, Egelsbach, Häsel-Hohenhausen, pages 193-202, 1997. 1.8.2, 1.8.3, 1.8.4, 1.13

[Ehrgott 00] M. Ehrgott, *Multicriteria Optimization*, Lecture Notes in Economics and Mathematical Systems, numéro 491, Springer Verlag, 2000. 1.8.7, 2.13, 9.2.1, 9.2.2

[Ehrgott et al. 00] M. Ehrgott, X. Gandibleux, *An Annotated Bibliography of Multiobjective Combinatorial Optimization*, Rapport Technique 62/2000, Fachbereich Mathematik, Université de Kaiserslautern, Germany, 2000. 9.2.3

[Elmohamed 97] S. Elmohamed, P. Coddington, G. Fox, *A Comparison of Annealing Techniques for Academic Course Scheduling*, Rapport technique NPAC SCCS-777, janvier 1997.

[Engrand 97] P. Engrand, *Approche d'optimisation multiobjectif basée sur l'adoucissement simulé, et son application à la gestion des combustibles nucléaires*, Icone 5 - Int. Conf. on Nuclear Engineering, volume 10, pages 1-8, 1997.

[Engrand et al. 98] P. Engrand, X. Mouney, *Une méthode originale d'optimisation multiobjectif*, Note interne EDF-DER numéro HT-14/97/035/A, mars 1998. 5.4.1.1

[Eschenauer et al. 90] H. Eschenauer, J. Koski, A. Osyczka, *Multicriteria design optimization : procedures and applications*, Springer-Verlag, 1990. 2.4.4, 3.3.1, 3.4.1, 3.5.1, 3.6.1, 3.8

[Ferber 95] J. Ferber, *Les systèmes multi-agents : vers une intelligence collective*, éditions InterEditions, 1995

[Fleming et al. 85] P. J. Fleming, A. P. Pashkevich, *Computer aided control system design using a multiobjective optimization approach*, Proceedings of the IEEE Control 85 Conference, pages 174-179, janvier 1985.

[Fleming 93] P. J. Fleming, *Genetic Algorithms for Multiobjective Optimization : Formulation, Discussion and Generalisation*, Genetic Algorithms : Proceedings of the Fifth International Conference, 1993. 11.4.2.4

[Flexer 99] A. Flexer, *On the Self-Organizing Maps for Clustering and Visualization*, Third European Conference on Principles of Data Mining and Knowledge Discovery, PKDD'99, Prague, Lecture notes in Artificial Intelligence, numéro 1704, Springer-Verlag, pages 80-88, 1999.

[Fonseca et al 93] C. M. Fonseca, P. J. Fleming, *Genetic Algorithms for Multiobjective Optimization : Formulation, Discussion and Generalization*, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993. 5.7.2.1.1

[Fudenberg et al. 92] D. Fudenberg, J. Tirole, *Game theory*, The MIT Press, 1992.

[Gandibleux 99] X. Gandibleux, *Journée de travail PM20 Programmation mathématique multiobjectif*, LAMIH-ROAD, Valenciennes, septembre 1999.

[Gandibleux 00] X. Gandibleux, *Journée de travail PM2O Programmation mathématique multiobjectif*, Tours, novembre 2000. 5.7.2.5

[Gandibleux 01] X. Gandibleux, *Journée de travail PM2O Programmation mathématique multiobjectif*, Paris, mai 2001.

[Gaudier 99] F. Gaudier, *Modélisation par réseaux de neurones, application à la gestion du combustible dans un réacteur*, Rapport CEA-R-5854, CEA/Saclay, 1999.

[Gert et al. 97] I. P. Gert, J. L. Underwood, *The Logic of Search Algorithms : Theory and Applications*, Constraint Programming Conference, CP97, 1997.

[Giesy 78] D. P. Giesy, *Calculation of Pareto optimal solutions to multiple objective problems using threshold of acceptability constraints*, IEEE Trans. on Autom. Contr., volume AC-23, numéro 6, décembre 1978. 2.12.1

[Glover et al.] F. Glover, M. Laguna, M. Marti, *Scatter Search*, Preprint.

[Glover 86] F. Glover, *Future paths for integer programming and links to artificial intelligence*, Comput. and Ops. Res., volume 13, numéro 5, pages 533-549, 1986. 5.5

[Glover 90] F. Glover, *Artificial intelligence, heuristic frameworks and tabu search*, Managerial and decision economics, volume 11, pages 365-375, 1990. 5.5

[Glover et al. 97] F. Glover, M. Laguna, *Tabu search*, Kluwer Academic Publishers, 1997. 5.5, 5.8

[Goicoechea et al. 79] A. Goicoechea, L. Duckstein, M. M. Fogel, *Multiple objective under uncertainty : An illustrative application of PROTRADE*, Water resources research, volume 15, numéro 2, pages 203-210, avril 1979.

[Goldberg 94] D. Goldberg, *Algorithmes génétiques*, éditions Addison Wesley, juin 1994. 5.6, 5.8

[Gomez-Skarmeta et al. 98] A. F. Gómez-Skarmeta, F. Jiménez, J. Ibáñez, *Pareto-optimality in Fuzzy Modeling*, In 6th European Congress on Intelligent Techniques and Soft Computing EUFIT'98, pages 694-700, septembre 1998.

[Guvenir et al. 95] H. A. Güvenir, *A Genetic Algorithm for Multicriteria Inventory Classification*, Artificial Neural Nets and Genetic Algorithms. Proceedings of the International Conference, pages 6-9, Vienne, Autriche, avril 1995.

[Haimes et al. 75] Y. Y. Haimes, W. A. Hall, H. T. Freedman, *Multiobjective optimization in water resources systems*, éditions Elsevier scientific publication company, 1975. 3.2.1

[Han et al. 01] Q. Han, J. Han, *Revised Filled Function Methods for Global Optimization*, Applied Math. and Comp., numéro 119, pages 217-228, 2001.

[Hansen 97a] M. P. Hansen, *Experiments on the usage of hashing vectors in multiobjective tabu search*, août 1997.

[Hansen 97b] M. P. Hansen, *Tabu search for multiobjective optimization : MOTS*, MCDM97, janvier 1997.

[Hansen 97c] M. P. Hansen, *Generating a Diversity of Good Solutions to a Practical Combinatorial Problem using Vectorized Simulated Annealing*, Preprint, 1997.

[Hansen et al. 98] P. Hansen, N. Mladenović, *Variable neighborhood search : Principles and applications*, Les cahiers du GERAD G-98-20, mai 1998.

[Hansmann et al. 94] U. H. E. Hansmann, Y. Okamoto, *Comparative Study of Multicanonical and Simulated Annealing Algorithms in the Protein Folding Problem*, Rapport Technique SC-94-20-NWU-5/94, juillet 1994.

[Hartmann 99] A. K. Hartmann, *Introduction to Complexity Theory*, octobre 1999

[Hati et al. 87] S. K. Hati, R. G. Lamb, *Application of game theory in the design of optimal air pollution control measures*, Atmospheric environment, volume 21, numéro 8, pages 1833-1841, 1987.

[Henrion 74] C. Henrion, *L'aide à la décision, état de l'art et des problèmes*, Informatiques et gestion, numéro 60, 1974.

[Hillier et al. 95] F. S. Hillier, G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill International Editions, 1995. (document), 1.13

[Hoffmann et al. 95] T. Hoffmann, J. Buhmann, *Multidimensional scaling and data clustering*, Advances in Neural Information Processing 7, Morgan Kaufmann Publishers, pages 104-111, 1995.

[Hong et al. 86] L. Hong, C. Ting, *Group decision making with multiple objectives and its interactive solution approach*, IFAC Large scale systems : Theory and applications, 1986.

[Hooker 95] J. N. Hooker, *Testing Heuristics : We Have It All Wrong*, Journal of Heuristics, pages 33-42, mai 1995. 7.16

[Hordijk] W. Hordijk, *A Measure of Landscapes*, Evolutionary Computation, volume 4, numéro 4, pages 335-360, 1996.

[Horn et al. 93] J. Horn, N. Nafpliotis, D. E. Goldberg, *Multiobjective optimization using the niched Pareto genetic algorithm*, Rapport technique, IlliGAl Report 93005, université de l'Illinois, Urbana, Illinois, 1993. 5.7.2.3.1

[James 95] R. James, *A Framework for Search Heuristics*, 1995.

[Jaszkiewicz 01] A. Jaszkiewicz, *Multiple Objective Metaheuristic Algorithms for Combinatorial Optimization*, Thèse d'habilitation, Poznan University of Technology, 2001.

[Jimenez et al. 99] F. Jiménez, J. L. Verdegay, A. F. Gómez-Skarmeta, *Evolutionary techniques for constrained multiobjective optimization problems*, Proceedings of the 1999 Genetic and Evolutionary Computation Conference, pages 115-116, Orlando, Floride, juillet 1999.

[Jin et al. 01] Y. Jin, T. Okabe, B. Sendhoff, *Adapting Weighted Aggregation for Multiobjective Evolution Strategies*, EMO 2001, pages 96-110, 2001.

[Jones et al. 95] T. Jones, S. Forrest, *Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms*, Proc. of the sixth Int. Conf. on GA, pages 184-192, 1995.

[Kahng et al. 97] A. B. Kahng, R. Sharma, *Studies of Clustering Objectives and Heuristics for Improved Standard-Cell Placement*, Manuscript, 1997

[Kanzow 98] C. Kanzow, *Global Optimization Techniques for Mixed Complementarity Problems*, Technical Report MP-TR-1998-09, juillet 1998.

[Keeney 82] R. L. Keeney, *Decision analysis : an overview*, Operations Research, volume 30, numéro 5, pages 803-837, septembre 1982

[Keeney et al. 93] R. L. Keeney, H. Raiffa, *Decisions with multiple objectives : preferences and value tradeoff*, éditions Cambridge University Press, 1993/ (document), 2.2.1

[Keller] P. M. Keller, *FORMOSA-P Constrained Multiobjective Simulated Methodology*

[Kennedy et al. 95] J. Kennedy, R. Eberhart, *Particle Swarm Optimization*, Proc. IEEE Int. Conf. on Neural Net., pages 1942-1948, 1995.

[Kermanshahi et al. 89] B. S. Kermanshahi, Y. Yasuda, R. Yokoyama, *Environmental marginal cost evaluation by non-inferiority surface for optimal power dispatching*, Rapport technique PE-89-177, 1989.

[Kim et al.] D. G. Kim, P. Husbands, *Mapping based constraint handling for evolutionary search ; Thurston's circle packing and grid generation*, Preprint.

[Kingdon et al. 95] J. Kingdon, L. Dekker, *The Shape of Space*, Proceedings of the first IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems : Innovations and Applications, IEE, London, pages 543-548, 1995.

[Klock et al. 97] H. Klock, J. M. Buhmann, *Multidimensional scaling by deterministic annealing*, Springer Lecture notes in computer science, numéro 1223, pages 246-260, mai 1997.

[Klock et al. 99] H. Klock, J. M. Buhmann, *Data visualization by multidimensional scaling : A deterministic annealing approach*, Pattern Recognition, volume 33, numéro 4, pages 651-669, 1999.

[Knowles et al. 99] J. Knowles, D. W. Corne, *The Pareto Archived Evolution Strategy : A new baseline algorithm for Pareto multiobjective optimization*, 1999 Congress on Evolutionary Computation, pages 98-105, juillet 1999.

[Knowles et al. 00] J. D. Knowles, D. W. Corne, *A comparison of Diverse Approaches to Memetic Multiobjective Combinatorial Optimization*, Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, pages 103-108, Las Vegas, Nevada, USA, juillet 2000.

[Knowles et al. 01] J. D. Knowles, R. A. Watson, D. W. Corne, *Reducing Local Optima in Single-Objective Problems by Multi-objectivization*,First International Conference on Evolutionary Multi-Criterion Optimization, pages 268-282, numéro 1993, 2001

[Koppen et al. 98] M. Köppen, S. Rudlof, *Multiobjective Optimization by Nessy Algorithm*, Proceedings of the 3rd Online Workshop on Soft Computing, pages 357-368, Londres, 1998.

[Koski 85] J. Koski, *Defectiveness of weighting method in multicriterion optimization of structures*, Comm. in Appl. Num. Math, volume 1, pages 333-337, 1985. 2.1.4

[Koza et al. 99] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, *Genetic Programming III : Automatic Programming and Automatic Circuit Synthesis*, Morgan Kauffman Publishers, mai 1999.

[Krishnamachari et al. 00] B. Krishnamachari, X. Xie, B. Selman, S. Wicker, *Performance Analysis of Neighborhood Search Algorithms*, Preprint, janvier 2000.

[Kruskal 64a] J. B. Kruskal, *Nonmetric multidimensional scaling : a numerical method*, Psychometrika, volume 29, numéro 2, juin 1964.

[Kruskal 64b] J. B. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, volume 29, numéro 1, mars 1964.

[Laguna et al. 99] M. Laguna, R. Marti, *GRASP and path relinking for 2-layer straight line crossing minimization*, INFORMS Journal on computing, volume 11, numéro 1, 1999.

[Lamagna 87] E. A. Lamagna, *Infeasible Computation : NP Complete Problems*, ABACUS, volume 4, numéro 3, pages 18-33, 1987.

[Laumanns et al. 00] M. Laumanns, E. Zitzler, L. Thiele, *A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism*, 2000 Congress on Evolutionary Computation, volume 1, pages 46-53, Piscataway, New Jersey, USA, juillet 2000. 5.7.2.5, 7.15

[Lauriere 78] J. L. Laurière, *A language for stating and solving combinatorial problems*, Artificial intelligence, numéro 10, pages 29-127, 1978

[Lee et al. 00] Y.-H. Lee, B. J. Berne, *Global Optimization : Quantum Thermal Annealing with Path Integral Monte Carlo*, J. Phys. Chem. A, volume 104, pages 86-95, 2000.

[De Leeuw 93] J. De Leeuw, *Fitting distances by least squares*, Rapport technique, UCLA Statistics Series 130, LA, 1993.

[De Leeuw et al. 97] J. De Leeuw, P. J. F. Groenen, *Inverse multidimensional scaling*, Journal of Classification, volume 14, pages 3-21, 1997.

[Levin] M. S. Levin, *Algorithm Systems for Combinatorial Optimization : Hierarchical Multistage Framework*, Preprint.

[Lin 76] J. G. Lin, *Multiple-objective problems : Pareto-optimal solutions by proper equality constraints*, IEEE Transactions on Automatic Control., volume 5, numéro AC21, pages 641-650, octobre 1976. 2.9.1

[Lin 77] J. G. Lin, *Proper inequality constraints and maximization of index vectors*, Journal of optimization theory and applications, volume 21, numéro 4, pages 505-521, avril 1977. 2.10.1

[Lin et al. 98] C. Lin, J. I. Yang, K. J. Lin, Z. D. Wang, *Pressurized water reactor loading pattern design using the simple tabu search*, Nuclear science and engineering, volume SC1, pages 61-71, 1998.

[Lister 93] R. Lister, *Annealing Networks and Fractal Landscapes*, IEEE Int. Conf. on Neural Networks, volume 1, pp 257-262, San Francisco, mars 1993.

[Liu 01] X. Liu, *A Class of Generalized Filled Functions with Improved Computability*, Journal of Comp. and Applied Math., numéro 137, pages 61-69, 2001.

[Liu 02] X. Liu, *A Computable Filled Function used for Global Minimization*, Applied Math. and Comp., numéro 126, pages 271-278, 2002.

[Lombard 79] J. Lombard, *Revue de quelques méthodes d'aide à la décision*, Commission des communautés Européennes, 2-85206-127-9, Technique et documentation (FRA), pages 69-88, 1979.

[Lootsma 87] F. A. Lootsma, *Modélisation du jugement humain dans l'analyse multicritère du moyen de comparaison par nos pairs*, R.A.I.R.O. Recherche opérationnelle, volume 21, numéro 3, pages 241-257, août 1987.

[Loukil et al. 00] T. Loukil, J. Teghem, Ph. Fortemps, *Solving Multi-objective Production Scheduling Problems with Tabu Search*, Control and Cybernetics, volume 29, numéro 3, pages 819-828, 2000.

[Ma et al. 94] J. Ma, J. E. Straub, *Simulated Annealing using the classical density distribution*, J. Chem. Phys, volume 101, numéro 1, juillet 1994.

[Mathias et al. 94] K. E. Mathias, L. D. Whitley, *Transforming the Search Space with Gray Coding*, IEEE Conf. on Ev. Comp., volume 1, pages 513-518, 1994.

[Mathieu et al. 96] P. Mathieu, J.-P. Delahaye, *Expériences sur le dilemme itéré des prisonniers*, Publication Interne du LIFL, numéro IT-233, mars 1996.

[Maystre et al. 94] L.-Y. Maystre, J. Pictet, J. Simos, *Méthodes multicritères ELECTRE*, édition Presses Polytechniques et Universitaires Romandes, 1994. 6.2.4, 6.3.1.2, 6.3.1.2, 6.3.2.1, 6.3.3.1, 6.3.4.4, 6.3.5.1, 6.3.5.2, 6.3.6.1, 6.3.7.1, 6.4

[McAllester et al. 97] D. McAllester, B. Selman, H. Kautz, *Evidence for Invariants in Local Search*, Proceedings of the Fourteenth National Conference on Artificial Intelligence, Providence, Rhode Island, pages 321-326, 1997.

[McReady et al. 95] W. G. McReady, D. H. Wolpert, *What makes an Optimization Problem Hard ?*, Rapport Technique, SFI-TR-95-0, Santa Fe Institute, avril 1995.

[Menon et al. 95] A. Menon, K. Mehrotra, C. K. Mohan, S. Ranka, *Optimization Using Replicators*, Proc. of the sixth Int. Conf. on GA, pages 209-216, 1995.

[Menon et al. 97] A. Menon, K. Mehrotra, C. K. Mohan, S. Ranka, *Replicators, Majorization and Genetic Algorithms : New Models and Analytical Tools*, Foundations of GA, numéro 4, 1997.

[Mertens] S. Mertens, *Computational Complexity for Physicists*, Notes de cours.

[Merz 00] P. Merz, *Memetic Algorithms for Combinatorial Optimization Problems : Fitness Landscape ans Effective Search Strategies*, Thèse, décembre 2000.

[Merz et al. 98] P. Merz, B. Freisleben, *On the Effectiveness of Evolutionary Search in High-Dimensional NK-Landscapes*, Proc. of the 1998 IEEE Int. Conf. on Ev. Comp., pages 741-745, 1998.

[Messac et al. 00] A. Messac, G. J. Sundararaj, R. V. Tappeta, J. E. Renaud, *Ability of Objective Functions to Generate Points on Nonconvex Pareto Frontiers*, AIAA Journal, volume 38, numéro 6, pages 1084-1091, juin 2000. 2.3.4

[Michalewicz 95a] Z. Michalewicz, *A survey of constraint handling techniques in evolutionary computation methods*, Proceedings of the Fourth Annual Conference on Evolutionary Programming, MIT Press, Cambridge, pages 135-155, 1995

[Michalewicz 95b] Z. Michalewicz, *Genetic algorithms, numerical optimization, and constraints*, Proceedings of the sixth International Conference on Genetic Algorithm, pages 151-158, San Mateo, Californie, USA, juillet 1995.

[Miettinen et al. 98] K. M. Miettinen, M. M. Mäkelä, *Proper Pareto optimality in nonconvex problems - characterization with tangent and normal cones*, Rapport technique, Université de Jyväskylä, novembre 1998.

[Miettinen 99] K. M. Miettinen, *Nonlinear multiobjective optimization*, éditions Kluwer academic publisher, 1999. 1.8.7, 1.13, 2.3.1, 2.4.1, 2.5.3, 2.7.3, 3.8, 9.1

[Mobius et al.] A. Mobius, A. Diaz-Sanchez, B. Freisleben, M. Schreiber, A. Fachat, K. H. Hoffmann, P. Merz, A. Neklioudov, *Two Physically Motivated Algorithms for Combinatorial Optimization : Thermal Cycling and Iterative Partial Transcription*.

[Monclar 95] F.-R. Monclar, *Résolution coopérative de problèmes : notes bibliographiques*, Note interne EDF-DER numéro HT-37/95/019, 1995

[Monclar 98] F.-R. Monclar, *Résolution coopérative de problèmes : ELICO et son application à la supervision des réseaux électriques*, Thèse de l'université de Montpellier (LIRMM), mars 1998.

[Monclar 99] F.-R. Monclar, *Deux exemples d'utilisation d'algorithmes "méta-heuristiques" pour des applications à EDF*, Note interne EDF-DER, 1999.

[MOPGP] M.O.P.G.P. Site Internet sur la méthode du but programmé. 2.7.3

[More et al. 95] J. J. Moré, Z. Wu, *Global Smoothing and Continuation for Large-Scale Molecular Optimization*, preprint MCS-P539-1095, octobre 1995.

[Morita et al. 01] H. Morita, X. Gandibleux, N. Katoh, *Experimental Feedback on Biobjective Permutation Scheduling problems Solved with a Population Heuristic*, FCDS, volume 26, numéro 1, pp 43-50, 2001.

[Mouney 98] X. Mouney, *Implémentation dans le logiciel FORMOSA d'une méthode multi-objectif basée sur le recuit simulé*, Rapport de stage EDF-DER, 1998.

[Muller 94] C. Muller, *Introduction aux méthodes neuronales d'optimisation*, Note interne EDF-DER numéro HI-23/94/012, décembre 1994.

[Multisimplex] Manuel d'utilisation du logiciel MultiSimplex 3.7.1, 3.7.3

[Munro et al. 86] J. Munro, P. H. Chaung, *Optimal plastic design with imprecise data*, Journal of engineering mechanics, volume 112, numéro 9, pages 888-903, septembre 1986. 4.2.2

[Mynard 97] L. Mynard, *Exploration locale oscillante heuristiquement ordonnée*, Thèse LAFORIA-LIP6, décembre 1997.

[Nelder et al. 65] J. A. Nelder, R. Mead, *A Simplex method for function minimization*, Computer Journal, volume 7, pages 308-313, 1965. 3.7.1

[Niimura et al.] T. Niimura, Y. Ueki, T. Matsumoto, R. Yokoyama, *A fuzzy approach for multi-objective optimal generation dispatch*, preprint. 4.2.2

[Norbis et al. 88] M. I. Norbis, J. McGregor-Smith, *A multiobjective, multi-level heuristic for dynamic resource constrained scheduling problems*, Eur. Journal of Op. Research, volume 33, numéro 1, pages 30-41, 1988.

[Opricovic] S. Opricovic *An extension of compromise programming to the solution of dynamic multicriteria problems*, preprint.

[Osyczka 92] A. Osyczka, *Computer Aided Multicriterion Optimization System (CAMOS)*, International Software Publisher, 1992.

[Othmani 98] I. Othmani, O*ptimisation multicritère : Fondements et Concepts*, Thèse de doctorat, université de Grenoble, mai 1998. 1.8.6

[Papadimitriou 97] C. H. Papadimitriou, *NP Completeness : A Retrospective*, Proceedings of ICALP 97, Springer, 1997.

[Parks et al. 98] G. T. Parks, I. Miller, *Selective Breeding in a Multiobjective Genetic Algorithm*, PPSN V, pages 250-259, Amsterdam, Holland, 1998. 5.7.2.5

[Parks et al. 99] G. T. Parks, A. Suppapitnarm, *Multiobjective Optimization of PWR Reload Core Designs using Simulated Annealing*, Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, Madrid, Espagne, pages 1435-1444, septembre 1999.

[Pasquier-Dorthe et al. 95] J. Pasquier-Dorthe, H. Raynaud, *Un outil d'aide à la décision multicritère*, Revue Française de Gestion, numéro 106, pages 11-21, 1995.

[Pentzaropoulos et al. 93] G. C. Pentzaropoulos, D. I. Giokas, *Cost-performance modeling and optimization of network flow balance via linear goal programming analysis*, Computer Communications, volume 16, numéro 10, pages 645-652, 1993. 2.6.2

[Peralska et al.] E. Peralska, D. De Ridder, R. P. W. Duin, M. A. Kraaijveld, *A new method of generalizing Sammon mapping with application to algorithm speed up*, Fourth Annual Conference of the Advanced School for Computing and Imaging, pages 221-228, juin 1999.

[Peretto] P. Peretto, *Réseaux de neurones et optimisation combinatoire*, preprint

[Perrin et al. 97] E. Perrin, A. Mandrille, M. Oumoun, C. Fonteix, I. Marc, *Optimisation globale par stratégie d'évolution : Technique utilisant la génétique des individus diploïdes*, Recherche Opérationnelle, AFCET-Gauthier-Villars, volume 31, numéro 2, pages 161-201, 1997. 10.4

[Peterson et al. 93] C. Peterson, B. Soderberg, *A new method for mapping optimization problems onto neural networks*, International journal of neural systems, volume 1, numéro 1, pages 3-22, 1993.

[Pohlheim] H. Pohlheim, *Visualization of Evolutionary Algorithms : Real-World Application of Standard Techniques and Multidimensional Visualization*, Preprint.

[Potter 02] M. A. Potter, *An NK-Landscape Generator*, 2002.

[Pouillot 92] S. Pouillot, *Etude comparative des techniques multi-agents et des blackboard*, Note Interne EDF numéro HI-57/7959, mai 1992.

[Press et al. 02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C++*, Cambridge University Press, 2002.

[Preux et al. 99] Ph. Preux, E.-G. Talbi, *Towards Hybrid Evolutionary Algorithms*, Int. Trans. in Op. Res., volume 6, pages 557-570, 1999.

[Quentin 82] R. Quentin, *Quatre techniques d'aide à la décision*, Cadréco, numéro 63, pages 9-12, septembre 1982.

[Radcliff et al. 95] N. J. Radcliff, P. D. Surry, *Fundamental Limitations on Search Algorithms : Evolutionary Computation in Perspective*, Lecture Notes in Computer Science 1000, Springer-Verlag, 1995.

[Reardon 97a] B. J. Reardon, *Fuzzy logic vs niched Pareto multi-objective genetic algorithm optimization : Part I : Schaffer's F2 problem*, Rapport technique LA-UR-97-3675, Los Alamos National Laboratory, 1997. 4.3.1, 4.3.3

[Reardon 97b] B. J. Reardon, *Fuzzy logic vs niched Pareto multi-objective genetic algorithm optimization : Part II : a simplified Born-Mayer problem*, Rapport technique LA-UR-97-3676, Los Alamos National Laboratory, 1997. 4.3.1, 4.3.3

[Rebreyend 99] P. Rebreyend, *Algorithmes génétiques hybrides en optimisation combinatoire*, Thèse de l'E.N.S. Lyon, janvier 1999.

[Reidys et al.] C. M. Reidys, P. F. Stadler, *Combinatorial Landscapes*.

[Remez 86] C. Remez, *L'intelligence dévoilée ou la logique floue*, Micro-systèmes, mai 1986. 4.1

[Reuss 85a] P. Reuss, *Eléments de neutronique*, Notes de cours I.N.S.T.N., 1985.

[Reuss 85b] P. Reuss, *Que sais-je ? La neutronique*, P.U.F., 1985.

[Robert 85] F. Robert, *Meilleure approximation en norme vectorielle et minimum de Pareto*, Math. Modelling and Num. Analysis, volume 19, numéro 1, pages 90-110, 1985.

[Roy 85] B. Roy, *Méthodologie multicritère d'aide à la décision*, éditions Economica, 1985.

[Roy et al. 81] B. Roy, P. Vincke, *Multicriteria analysis : survey and new directions*, European journal of Op. Research, 1981.

[Roy et al. 88] B. Roy, D. Bouyssou, *Aide à la décision*, Encyclopédie des sciences de la gestion (AFCET/Interfaces), numéro 65, pages 4-13, mars 1988.

[Roy et al. 93] B. Roy et D. Bouyssou, *Aide multicritère à la décision : Méthodes et cas*, éditions Economica, 1993. (document), 6.3.5.2, 6.3.5.2, 6.4

[Rudolph 98] G. Rudolph, *On a Multiobjective Evolutionary Algorithm and its Convergence to the Pareto Set*, Proceedings of the 5th IEEE Conference on Evolutionary Computation, pages 511-516, Piscataway, New Jersey, USA, 1998.

[Sammon 69] J. M. Sammon, *A nonlinear mapping for data structure analysis*, IEEE Transactions on Computers, volume C-18, numéro 5, pages 401-409, 1969

[Sait et al. 99] S. M. Sait, H. Youssef, *Iterative Computer Algorithms with Applications in Engineering : Solving Combinatorial Optimization Problems*, IEEE Computer Society, 1999. 5.1, 5.8

[Sakawa et al. 77] N. Sakawa, Y. Sawaragi, *Multiobjective optimization in water resources problem for a single river basin - A case study of the Kakogawa river basin*, Proceedings of the First KNKI IRDP Workshop, pages 197-221, 1977.

[Sakawa et al. 85] M. Sakawa, F. Seo, *Interactive multiobjective decision making by the sequential proxy optimization technique (SPOT) and its application to environmental systems*, I.F.A.C. control science and technology, volume 10, numéro 50-2, pages 119-124, 1985.

[Sakawa et al. 88] M. Sakawa, H. Yano, *An interactive fuzzy satisficing method for multiobjective linear programming problems with fuzzy parameters*, Fuzzy Sets and Systems, volume 28, pages 129-144, 1988. 4.2.1

[Sakawa 02] M. Sakawa, *Genetic Algorithms and Fuzzy Multiobjective Optimization*, Kluwer Academic Publishers, 2002. 4.2.1

[Sarma et al. 96] J. Sarma, K. De Jong, *An Analysis of the Effects of Neighborhood Size and Shape on Local Selection Algorithms*, In Proc. 4th PPSN, LNCS 1141, Springer-Verlag, pages 236-244, 1996 .

[Sawaragi et al.] Y. Sawaragi, S. Ikeda, H. Nakayama, T. Tanino, C. Okumura, *An interactive optimization method for a water resource problem with multiple objectives*, preprint.

[Schaerf et al. 99] A. Schaerf, M. Lenzerini, M. Cadoli, *LOCAL++ : A C++ Framework for Local Search Algorithms*, Rapport Technique 11-99, Dipartimento di Informatica e Sistemistica, Università di Roma, Italie, mai 1999.

[Schaffer et al. 89] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, R. Das, *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*, ICGA-89, pages 51-60, San Mateo, Californie, USA, 1989.

[Scharlig 85] A. Scharlig, *Décider sur plusieurs critères : panorama de l'aide à la décision multicritère*, éditions Presses Polytechniques Romandes, 1985. 6.1, 6.4

[Scharlig 96] A. Scharlig, *Pratiquer ELECTRE et PROMETHEE : Un complément à Décider sur plusieurs critères*, éditions Presses Polytechniques Romandes, 1996. 6.4

[Schelstraete et al.] S. Schelstraete, W. Schepens, H. Verschelde, *Energy minimization by smoothing techniques : a survey*.

[Schoenauer et al. 98] M. Schoenauer, Z. Michalewicz, *Sphere operators and their applicability for constrained parameter optimization problems*, Seventh Annual Conference on Evolutionary Programming - EP'98, 1998.

[Sefrioui et al.] M. Sefrioui, J. Periaux, *Nash Genetic Algorithms : Examples and Applications*, Preprint.

[Sen] S. Sen, *Stochastic programming : Computational issues and challenges*, Encyclopedia of OR/MS.

[Serquin et al. 97] Y. Serquin, F. Beaudouin, B. Murier, *Multi-attribute utility theory toward a more general framework*, Note interne EDF-DER numéro HP-28/97/042/A, avril 1997.

[Shao et al. 95] C.-S. Shao, R. H. Byrd, E. Eskow, R. B. Schnabel, *Global Optimization for Molecular Clusters using a new Smoothing Approach*, 1995.

[Sharpe 00] O. J. Sharpe, *Towards a Rational Methodology for Using Evolutionary Search Algorithms*, Thèse, University of Sussex, England, janvier 2000.

[Shatilla et al. 99] Y. A. Shatilla, D. C. Little, J. A. Penkrot, R. A. Holland, *Biased multi-objective function optimization in Westinghouse advanced loading pattern search code ALPS*, Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, volume 2, pages 282-295, Madrid, Espagne, septembre 1999.

[Siarry et al. 89] P. Siarry, G. Dreyfus, *La méthode du recuit simulé. Théorie et applications*, éditions IDSET, ESPCI, 1989.

[Siarry 94] P. Siarry, *La méthode du recuit simulé en électronique*, Rapport d'habilitation à diriger les recherches. Université de Paris-Sud (Orsay), 1994. 5.4

[Siarry 99] P. Siarry, *Optimisation et classification de données*, Cours du D.E.A G.B.M. de l'université de Paris 12, 1999. 5.4

[Smith 95] B. M. Smith, *A tutorial on constraint programming*, Rapport technique 95-14, avril 1995.

[Smith 99] K. A. Smith, *Neural networks for combinatorial optimization : A review for more than a decade of research*, INFORMS Journal on Computing, volume 11, numéro 1, 1999.

[Spenle et al. 96] D. Spenlé. R. Gourhant, *Guide du calcul en mécanique*, Hachette Technique, 1996. 1.6.3.1

[Spierenburg 97] J. A. Spierenburg, *Dimension Reduction of Images using Neural Networks*, Master Thesis, Department of Computer Science, Leiden University, juillet 1997.

[Srinivas et al. 93] N. Srinivas, K. Deb, *Multiobjective optimization using non-dominated sorting in genetic algorithms*, Rapport technique, Department of Technical Engineering, Indian Institute of Technology, Kanput, India, 1993. 5.7.2.2.1

[Stadler et al. 92] P. F. Stadler, W. Schnabl, *The Landscape of the Travelling Salesman Problem*, 1992.

[Steuer 76] R. E. Steuer, *Multiple objective linear programming with interval criterion weights*, Management science, volume 23, numéro 3, pages 305-316, novembre 1976.

[Stutzle et al. 00] T. Stutzle, A. Grun, S. Linke, M. Ruttger, *A Comparison of Nature Inspired Heuristics on the Traveling Salesman Problem*, 2000.

[Szidarowsky et al. 84] F. Szidarowsky, L. Duckstein, I. Bogardi, *Multiobjective management of mining under water hazard by game theory*, Journal of Op. Research, volume 15, pages 251-258, 1984.

[Tabak 79] D. Tabak, *Computer based experimentation with multicriteria optimization problems*, IEEE Transactions on Systems Man and Cybernetics, volume SMC9, numéro 10, pages 676-679, septembre 1979. 2.11.1

[Tagami et al. 99] T. Tagami, T. Kawabe, *Genetic Algorithm based on a Pareto Neighborhood Search for Multiobjective Optimization*, Proc. of the 1999 Int. Symp. of Nonlinear Th. and its Appl., pages 331-334, 1999.

[Taillard] E. D. Taillard, *Comparison of Non-Deterministic Iterative Methods*, preprint.

[Talbi 98] E.-G. Talbi, *A Taxonomy of Hybrid Metaheuristics*, Rapport technique TR-183, LIFL, Université de Lille 1, mai 1998. 9.1

[Tarvainen 86] K. Tarvainen, *On the generating of Pareto optimal alternatives in large scale systems*, IFAC Large scale systems : Theory and applications, Pergamon Press, pages 519-524, août 1986.

[Teghem et al. 00] J. Teghem, D. Tuyttens, E. L. Ulungu, *An Interactive Heuristic Method for Multi-objective Combinatorial Optimization*, Computer & Operations Research, volume 27, pages 621-634, 2000.

[Thiollet 99] B. Thiollet, *Méthode du gradient discret pour l'optimisation de la recherche de plans de rechargement : recommandations pour la maquette*, Note interne EDF-DER numéro HI-72/99/002/A, février 1999.

[Tipping et al. 97] M. E. Tipping, C. M. Bishop, *Mixtures of Principal Component Analysis*, Rapport technique, NCRG/97/003, Neural Computing Research Group, Université d'Aston, Birmingham, UK, 1997.

[Tipping et al. 98] M. E. Tipping, D. Toux, *Shadow Targets : A Novel Algorithm for Topographic Projections by Radial Basis Functions*, NeuroComputing, volume 19, pages 211-222, 1998.

[T'kindt et al. 00] V. T'kindt, J.-C. Billaut, *L'ordonnancement multicritère*, Presses Universitaires de Tours, décembre 2000.

[Toshinsky et al. 00] V. G. Toshinsky, H. Sekimoto, G. I. Toshinsky, *A method to improve multiobjective genetic algorithm optimization of a self-fuel-providing LMFBR by niche induction among nondominated solutions*, Annals of nuclear energy, volume 25, numéro 5, pages 397-410, 2000.

[Turinsky 99] P. J. Turinsky, *Mathematical optimization of incore Nuclear fuel management Decisions : status and Trends*, ATW. Atomwirtschaft-Atomtechnik - Internationale Zeitschrift Fur Kernenergie, volume 44, numéro 7, pages 454-458, 1999.

[Tuyttens et al. 00] D. Tuyttens, J. Teghem, Ph. Fortemps, K. Van Nieuwenhuyze, *Performance of the MOSA Method for the Bicriteria Assignment Problem*, Journal of Heuristics, volume 6, numéro 3, pages 295-310, 2000.

[Ulungu et al. 94] E. L. Ulungu, J. Teghem, *Multi-objective Combinatorial Optimization Problems : A Survey*, Journal of Multi-criteria Decision Analysis, volume 3, pages 83-104, 1994.

[Ulungu et al. 95] E. L. Ulungu, J. Teghem, *The Two Phases Method : An Efficient Procedure to Solve Bi-objective Combinatorial Optimization Problems*, Foundations of Computing and Decision Sciences, volume 20, numéro 2, 1995.

[Ulungu et al. 99] E. L. Ulungu, J. Teghem, P. H. Fortemps, D. Tuyttens, *MOSA Method : A Tool for Solving Multiobjective Combinatorial Optimization Problems*, Journal of Multicriteria Decision Analysis, volume 8, numéro 4, pages 221-236, 1999. 5.4.2.1, 5.4.2.3

[Vaessens et al. 92] R. J. M. Vaessens, E. H. L. Aarts, J. K. Lenstra, *A Local Search Template*, PPSN 2, 1992.

[Vallin et al. 00] P. Vallin, D. Vanderpooten, *Aide à la décision : Une approche par les cas*, Ellipses, 2000.

[Van Den Bergh et al.] F. Van Den Bergh, A. P. EngelBrecht, *Effects of Swarm Size on Cooperativee Particle Swarm Optimizers*.

[Van Den Bergh et al. 00] F. Van Den Bergh, A. P. Engelbrecht, *Cooperative Learning in Neural Networks using Particle Swarm Optimizers*, South African Computer Journal, numéro 26, pages 84-90, 2000.

[Van Hentenryck et al. 96] P. Van Hentenryck, D. McAllester, D. Kapur, *Solving polynomial systems using a branch and prune approach*, SIAM Journal of Numerical Analysis, volume 34, numéro 2, 1996.

[Van Kemenade 98] C. H. M. Van Kemenade, *Analysis of NK-XOR Landscape*, 1998.

[Van Laarhoven et al. 92] P. J. M. Van Laarhoven, E. H. L. Aarts, *Simulated Annealing : Theory and Applications*, Kluwer Academic Publishers, 1992.

[Van Veldhuizen et al. 98] D. A. Van Veldhuizen, G. B. Lamont, *Multiobjective evolutionary algorithm research : A history and analysis*, Rapport technique TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, Ohio, USA, octobre 1998.

[Van Veldhuizen 99] D. A. Van Veldhuizen, *Multiobjective evolutionary algorithms : classifications, analyses and new innovations*, Ph. D., Graduate School of Engineering. Air Force Institute of Technology, Wright Patterson AFB, Ohio, USA, janvier 1999. 1.12, 7.1, 7.10.1, 7.16

[Van Veldhuizen et al. 00] D. A. Van Veldhuizen, G. B. Lamont, *On Measuring Multiobjective Evolutionary Algorithms Performances*, 2000 Congress on Evolutionary Computation, volume 1, pages 205-211, Piscataway, New Jersey, USA, juillet 2000.

[Varanelli 96] J. M. Varanelli, *On the Acceleration of Simulated Annealing*, Thèse, Faculty of the School of Engineering and Applied Sciences, University of Virginia, mai 1996.

[Vavasis] S. A. Vavasis, *Complexity Issues in Global Optimization : A Survey*, Preprint

[Vern 92] J. L. Vern, *La logique floue : concepts et définitions*, Electronique Radio Plans, décembre 1992 4.1

[Vincke 89] Ph. Vincke, *L'aide multicritère à la décision*, éditions Ellipses, éditions de l'Université de Bruxelles, 1989 6.2.3, 6.3.2.1, 6.3.4.4, 6.3.5.1, 6.3.6.1

[Visee et al. 98] M. Visée, J. Teghem, M. Pirlot, E. L. Ulungu, *Two-phases Method and Branch and Bound Procedures to Solve the Bi-objective Knapsack Problem*, Journal of Global Optimization, volume 12, pages 139-155, 1998

[Weidl 96] J. Weidl, *The standard template library tutorial*, juillet 1996

[Wendt 97] O. Wendt, W. Konig, *Cooperative Simulated Annealing ; How Much Cooperation is Enough ?*, Rapport Technique, 1997.

[Wenzel et al.] W. Wenzel, K. Hamacher, *A Stochastic Tunneling Approach for Global Minimization of Complex Potential Energy Landscapes*.

[De Werra 90] D. De Werra, *Heuristics for graph coloring*, Computing Suppl., volume 7, pages 191-208, 1990. 5.5

[Westhoff et al. 96] F. H. Westhoff, B. V. Yarbrough, R. M. Yarbrough, *Complexity, Organisation, and Stuart Kauffman's The Origin of Order*, J. of Economic Behavior and Organization, volume 29, pages 1-25, 1996.

[Whales et al.] D. J. Whales, J. P. K. Doye, *Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms*.

[Whitley 91] L. D. Whitley, *Fundamental Principles of Deception in Genetic Search*, Foundations of GA, pages 222-241, 1991.

[Whitley et al. 95] L. D. Whitley, K. Mathias, S. Rana, J. Dzubera, *Building Better Test Functions*, Int. Conf. on GA, 1995.

[Whitley et al. 96] D. Whitley, K. Mathias, S. Rana, J. Dzubera, *Evaluating Evolutionary Algorithms*, Journal of Artificial Intelligence, volume 85, numéros 1-2, pages 245-276, 1996. 8.4

[Williamson 99] D. P. Williamson, *IBM Research Report : Lecture Notes on Approximation Algorithms Fall 1998*, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, New York, Rapport Interne RC 21409, février 1999.

[Wolpert et al. 95] D. H. Wolpert, W. G. McReady, *No Free Lunch Theorem for Search*, Rapport technique SFI-TR-95-02-010, Santa Fe Institute, février 1995.

[Wolpert et al. 96] D. H. Wolpert, W. G. McReady, *No Free Lunch Theorem for Optimization*, IEEE Trans. on Evolutionary Computation, avril 1996.

[Wong et al. 98] H. I. Wong, S. Crouzet, *Optimisation des plans de rechargement nucléaire*, Note Interne EDF-DER numéro HI-14/98/034/A, décembre 1998.

[Wu et al. 01] J. Wu, S. Azarm, *Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set*, Trans. of the ASME, volume 123, pages 18-25, mars 2001.

[Xu et al. 98] J. Xu, S. Y. Chiu, F. Glover, *Fine-Tuning a Tabu Search Algorithm with Statistical Tests*, Int. Trans. Op. Res., volume 5, numéro 3, pages 233-244, 1998.

[Xu et al. 99] H. Xu, B. J. Berne, *Multicanonical jump walk annealing : An efficient method for geometric optimization*, Journal of Chem. Phys., pages 299-306, 1999.

[Yamamoto et al. 99] A. Yamamoto, H. Hashimoto, *Applications of Temperature Parallel Simulated Annealing to Loading Pattern Optimizations of Pressurized Water Reactors*, Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, volume 2, pages 1445-1458, Madrid, Espagne, septembre 1999.

[Yao 96] X. Yao, *An Overview of Evolutionary Computation*, Chinese Journal of Advanced Software Research, volume 3, numéro 1, pages 12-29, 1996.

[Zadeh 65] L. A. Zadeh, *Fuzzy Sets*, Information and Control, volume 8, pages 338-353, 1965. 4.1

[Zionts et al. 76] S. Zionts, J. Wallenius, *An interactive programming method for solving the multiple criteria problem*, Management science, volume 22, numéro 6, pages 652-653, février 1976.

[Zitzler et al. 98] E. Zitzler, L. Thiele, *An Evolutionary Algorithm for Multiobjective Optimization : the Strength Pareto Approach*, Rapport Technique 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology, mai 1998. 5.7.2.5

[Zitzler et al. 99] E. Zitzler, K. Deb, L. Thiele, *Comparison of Multiobjective Evolutionary Algorithms : Empirical Results*, Rapport Technique 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, décembre 1999. 7.14, 7.14.1, 7.14.2, 7.16

Index

Index